日　本　国　特　許　庁
JAPAN　PATENT　OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application:　２００３年　６月　２日

出願番号
Application Number:　特願２００３－１５７２５５

[ST.10/C]:　[ＪＰ２００３－１５７２５５]

出願人
Applicant(s):　松下電器産業株式会社

２００３年　６月２０日

特許庁長官
Commissioner,
Japan Patent Office　太田信一郎

出証番号　出証特２００３－３０４８７２５

【書類名】　特許願

【整理番号】　2032740094

【提出日】　平成15年 6月 2日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F　9/00

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　浅井　理恵子

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　庄田　幸恵

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　廣田　照人

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　井藤　好克

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　佐藤　太一

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　松島　秀樹

【発明者】

【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内

【氏名】　阿部　敏久

【特許出願人】

【識別番号】　000005821

【氏名又は名称】　松下電器産業株式会社

【代理人】

【識別番号】　100090446

【弁理士】

【氏名又は名称】　中島　司朗

【先の出願に基づく優先権主張】

【出願番号】　特願2002-225289

【出願日】　平成14年 8月 1日

【手数料の表示】

【予納台帳番号】　014823

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9003742

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　暗号化データを復号して実行用メモリ空間に配置する装置、およびその方法

【特許請求の範囲】

【請求項１】　暗号化された形で保存されているプログラムをコンピュータシステム上での実行のために復号する暗号化データ復号装置であって、

プログラムを暗号化された複数の部分プログラムの集合の形で保持している記憶手段と、

前記部分プログラムの各々について、復号のタイミングを示すタイミング情報と、復号後の前記実行用メモリ空間における配置エリアを示す位置情報とからなるメモリ配置情報を、前記プログラムの暗号化の際に予め生成しておくメモリ配置情報生成手段と、

前記記憶手段から、前記タイミング情報に従って部分プログラムを順次読み出して復号する復号手段と、

前記復号手段によって復号された部分プログラムを、前記位置情報に従って前記実行用メモリ空間内の配置エリアに配置するロード手段と、を有すること、

を特徴とする暗号化データ復号装置。

【請求項２】　前記メモリ配置情報における位置情報は、前記複数の部分プログラムの少なくとも一部について、当該部分プログラムの復号より前に別の部分プログラムが配置されていた配置エリアに上書きされるような内容となっていること、

を特徴とする請求項１に記載の暗号化データ復号装置。

【請求項３】　前記メモリ配置情報のうち前記位置情報は暗号化された状態で保持されており、

前記ロード手段は、前記位置情報を復号し、復号後の位置情報が示す配置エリアに復号後の部分プログラムを配置すること、

を特徴とする請求項１又は２に記載の暗号化データ復号装置。

【請求項４】　前記ロード手段は、配置エリアに部分プログラムをロードする際、当該配置エリアのサイズが当該部分プログラムのサイズより大きければ、

サイズ差によって生じる当該配置エリアの空き領域にダミーデータを書き込むこと、

を特徴とする請求項１乃至３のいずれかに記載の暗号化データ復号装置。

【請求項５】　前記ロード手段は、ある配置エリアにロードされた部分プログラムの実行が終了した時点から所定時間が経過しても当該配置エリアへの次の部分プログラムの配置が行われない場合、当該配置エリア内に配置されている部分プログラムを消去すること、

を特徴とする請求項１乃至４のいずれかに記載の暗号化データ復号装置。

【請求項６】　暗号化処理の際に予め、前記複数の部分プログラムの少なくとも一部に、他の部分プログラムの復号処理に用いられる暗号鍵の一部又は全部を埋め込んでおく鍵埋め込み手段をさらに有し、

前記復号手段は、部分プログラムを復号する場合、それ以前に復号されて前記実行用メモリ空間に配置されている１個以上の他の部分プログラムに埋め込まれた暗号鍵を取得して当該復号対象の部分プログラムを復号すること、

を特徴とする請求項１乃至５のいずれかに記載の暗号化データ復号装置。

【請求項７】　暗号化処理の際に予め、前記複数の部分プログラムの少なくとも一部に、他の部分プログラムの復号処理に用いられる暗号鍵の全部又は一部を生成するための暗号鍵生成プログラムを埋め込んでおくプログラム埋め込み手段をさらに有し、

前記復号手段は、部分プログラムを復号する場合、それ以前に復号されて前記実行用メモリ空間に配置されている１個以上の他の部分プログラムに埋め込まれていた暗号鍵生成プログラムの実行によって生成された暗号鍵を用いて当該復号対象の部分プログラムを復号すること、

を特徴とする請求項１乃至５のいずれかに記載の暗号化データ復号装置。

【請求項８】　暗号化処理の際に予め、前記複数の部分プログラムの少なくとも一部に、他の部分プログラムの復号処理に用いられる暗号鍵を復号するための鍵用暗号鍵を埋め込んでおく鍵用暗号鍵埋め込み手段をさらに有し、

前記復号手段は、部分プログラムを復号する場合、それ以前に復号されて前記実行用メモリ空間に配置されている他の部分プログラムから取得した鍵用暗号鍵

を用いて復号した暗号鍵によって当該復号対象の部分プログラムを復号すること、

を特徴とする請求項１乃至５のいずれかに記載の暗号化データ復号装置。

【請求項９】　前記ロード手段は、最初に復号される部分プログラムの実行用メモリ空間への配置に先立って前記配置エリアの絶対アドレスを動的に決定すること、

を特徴とする請求項１乃至８のいずれかに記載の暗号化データ復号装置。

【請求項１０】　前記復号手段による復号処理は復号支援プログラムを用いて行われ、

前記復号支援プログラムの正当性を確認する復号プログラム確認手段を更に有し、

前記復号手段は、部分プログラムの復号に先立って、前記復号支援プログラム認証部に前記復号支援プログラムの正当性確認を行わせ、正当性が認証された場合にのみ部分プログラムの復号を行うこと、

を特徴とする請求項１乃至９のいずれかに記載の暗号化データ復号装置。

【請求項１１】　割込みが検知された場合に、不正アクセス防止処理として、前記実行用メモリ空間上に配置済みの部分プログラムを消去する不正アクセス防止手段を更に有すること、

を特徴とする請求項１乃至１０のいずれかに記載の暗号化データ復号装置。

【請求項１２】　前記不正アクセス防止手段は、前記不正アクセス防止処理実行の際に、ダミープログラムを実行させること、

を特徴とする請求項１１記載の暗号化データ復号装置。

【請求項１３】　前記不正アクセス防止手段は、正当なプログラム逆解析のための割込み発生位置の登録を予め受け付けておき、検知された割込みが当該登録された割込み発生位置で発生していた場合は、前記不正アクセス防止処理を実行しないこと、

を特徴とする請求項１１又は１２に記載の暗号化データ復号装置。

【請求項１４】　前記部分プログラムの各々について、前記記憶手段における格納位置を示す格納位置情報を暗号化した状態で保持する格納位置情報保持部

を更に有し、

前記復号手段は、前記格納位置情報保持部から読み出して復号した格納位置情報に従って、前記記憶手段から部分プログラムを読み出し、復号すること、

を特徴とする請求項１乃至１３のいずれかに記載の暗号化データ復号装置。

【請求項１５】　前記格納位置情報が正当なものか否かを判定する格納位置情報認証部を更に有し、

前記復号手段は、前記格納位置情報認証部によって格納位置情報が正当と判定された場合に、前記記憶手段から部分プログラムを読み出し、復号すること、

を特徴とする請求項１４に記載の暗号化データ復号装置。

【請求項１６】　コンピュータシステム上での実行を前提としてプログラムを暗号化する暗号化プログラム生成装置であって、

前記プログラムを複数の部分プログラムの単位で実行メモリ上に配置するために、前記部分プログラムの各々について、復号処理のタイミングを示すタイミング情報と、復号後の前記実行用メモリ空間における配置エリアを示す位置情報とからなるメモリ配置情報を生成するメモリ配置情報生成手段と、

前記プログラムを複数の部分プログラムの集合の形で暗号化するプログラム暗号化手段と、を有し、

前記メモリ配置情報生成手段は、前記複数の部分プログラムの少なくとも一部が、当該部分プログラムの復号より前に別の部分プログラムが配置されていた配置エリアに上書きされるように、秘匿性を優先して前記メモリ配置情報の内容を決定すること、

を特徴とする暗号化プログラム生成装置。

【請求項１７】　前記プログラム暗号化手段は、前記複数の部分プログラムの少なくとも一部に対し、暗号化処理に先立って、他の部分プログラムの復号処理に用いられる暗号鍵そのもの、又は当該暗号鍵の取得に必要なデータを埋め込み、

部分プログラムが復号される際には、それ以前に復号された他の部分プログラムに埋め込まれていた前記暗号鍵または前記データを用いて取得された暗号鍵が用いられること、

を特徴とする請求項１６に記載の暗号化プログラム生成装置。

【請求項１８】　暗号化された形で保存されているプログラムをコンピュータシステム上での実行のために復号する暗号化データ復号方法であって、

プログラムを暗号化された複数の部分プログラムの集合の形で記憶装置に格納する格納ステップと、

前記部分プログラムの各々について、復号のタイミングを示すタイミング情報と、復号後の前記実行用メモリ空間における配置エリアを示す位置情報とからなるメモリ配置情報を、前記プログラムの暗号化の際に予め生成しておくメモリ配置情報生成ステップと、

前記タイミング情報に従って、復号対象の部分プログラムを前記記憶装置から読み出して復号する復号ステップと、

前記復号ステップにおいて復号された部分プログラムを、前記位置情報に従って前記実行用メモリ空間内の配置エリアに配置するロードステップと、を有すること、

を特徴とする暗号化データ復号方法。

【請求項１９】　暗号化された形で保存されているプログラムをコンピュータシステム上での実行のために復号する暗号化データ復号装置としてコンピュータを動作させるためのプログラムであって、

前記暗号化データ復号装置は、

プログラムを暗号化された複数の部分プログラムの集合の形で保持している記憶手段と、

前記部分プログラムの各々について、復号のタイミングを示すタイミング情報と、復号後の前記実行用メモリ空間における配置エリアを示す位置情報とからなるメモリ配置情報を、前記プログラムの暗号化の際に予め生成しておくメモリ配置情報生成手段と、

前記記憶手段から、前記タイミング情報に従って部分プログラムを順次読み出して復号する復号手段と、

前記復号手段によって復号された部分プログラムを、前記位置情報に従って前記実行用メモリ空間内の配置エリアに配置するロード手段と、を有すること、

を特徴とするプログラム。

【発明の詳細な説明】

【０００１】

【発明が属する技術分野】

本発明は、暗号化されたデータ、特にプログラムを実行又は使用するに当たって復号する暗号化データ復号装置であり、さらにいえば、機密保護を実現しながら復号を行う装置、また、そうした復号方法に関する。

【０００２】

【従来の技術】

従来、機密保護などの目的で暗号化されたデータやプログラムを、コンピュータシステム上での再生又は実行のために復号する場合は、コンピュータシステム上で復号支援プログラムを実行させる。しかし、仕様が公開されたオープンなコンピュータシステムに置いては、プログラムの解析および改変が容易である。そのため、復号支援プログラムを改変して、本来参照や改変が許されないはずの復号後プログラムを参照・改変が可能な状態にする、という不正行為も可能となってしまう。

【０００３】

そこで、機密保護の強化のために、復号支援プログラム自体を暗号化しておき、復号処理時には復号支援プログラムを復号してデータ復号処理を行わせる方式（特開平９－６２３２：特許文献１）や、復号処理開始前に復号支援プログラムの正当性をチェックする方式（特開平１１－３９１５６：特許文献２）が提案されている。これらは、復号支援プログラムの改変を防止することで、復号後のプログラムやデータの機密を保護しようというものである。

【０００４】

【特許文献１】

特開平９－６２３２号公報

【０００５】

【特許文献２】

特開平１１－３９１５６号公報

【０００６】

【発明が解決しようとする課題】

しかしながら、正当な復号支援プログラムを実行して復号処理を行った場合でも、復号されたプログラムやデータがコンピュータシステムのメモリにロードされた時点で、不正な割り込み等により制御が奪われると、ロードされているプログラムやデータは参照や改変が可能となり、機密を保護できなくなる。

【０００７】

本発明は上記課題に鑑み、暗号化プログラムやデータの復号から実行の過程における機密性を高める暗号化データ復号装置及び復号方法、更に言えば、復号されてメモリにロードされた状態にあるプログラムやデータについて、不正な参照や改変を防止したり、不正参照される内容を最小限に抑えたりすることによって機密保護を実現する暗号化データ復号装置及び復号方法を提供することを目的とする。

【０００８】

【課題を解決するための手段】

上記の目的を達成するために、本発明は、暗号化された形で保存されているプログラムをコンピュータシステム上での実行のために復号する暗号化データ復号装置であって、プログラムを暗号化された複数の部分プログラムの集合の形で保持している記憶手段と、前記部分プログラムの各々について、復号のタイミングを示すタイミング情報と、復号後の前記実行用メモリ空間における配置エリアを示す位置情報とからなるメモリ配置情報を、前記プログラムの暗号化の際に予め生成しておくメモリ配置情報生成手段と、前記記憶手段から前記タイミング情報に従って部分プログラムを順次読み出して復号する復号手段と、前記復号手段によって復号された部分プログラムを、前記位置情報に従って前記実行用メモリ空間内の配置エリアに配置するロード手段と、を有すること、を特徴とする暗号化データ復号装置を提供する。

【０００９】

このような暗号化データ復号装置によれば、実行用メモリに配置されるのはプログラムの一部のみであるため、割り込みなどを利用してメモリ上のデータを不

正に参照される事態になったとしても、被害は最小限に抑制できる。また、前記メモリ配置情報における位置情報を、前記複数の部分プログラムの少なくとも一部について、当該部分プログラムの復号より前に別の部分プログラムが配置されていた配置エリアに上書きされるような内容とすれば、特定の部分プログラムが実行用メモリ上に存在する時間が短くなり、その分、不正参照されにくくなる。

【００１０】

また、上に述べた目的は、上記の特徴を有する暗号化データ復号装置が実行する暗号化データ復号方法や、当該復号方法をコンピュータに実行させるプログラムによっても達成することができる。

【００１１】

【発明の実施の形態】

以下、本発明に関わる暗号化データ復号装置の実施の形態について、図面を参照しながら説明する。

（概要）

図１は、本発明に関わる暗号化データ復号装置の実施の形態であるプログラムローダ１とその関連装置との構成を、これらが動作するコンピュータシステム上の他装置と共に示すブロック図である。なお、本実施の形態において、暗号化データ復号装置が復号する暗号化データはプログラムとする。

【００１２】

本実施の形態におけるプログラムローダ１は、オペレーションシステム（以下、「ＯＳ」）からの指示に応じて、２次記憶装置Ｓから暗号化プログラムを読み出して復号したうえで、実行用の共有メモリＭ上にロードする。「ロード」とは、プログラムを実行のためにメモリ空間に配置することである。

プログラムローダ１の特徴は、本来はメモリ管理の手法であるオーバーレイを暗号化プログラムの機密保護の用途に用いる点である。オーバーレイは、プログラムサイズよりも小さいメモリにプログラムを収めるために使われる手法であり、プログラムをセグメント（以下、「部分プログラム」という）に分割しておいて、同じメモリ領域に時間をずらして配置する、というものである。本実施の形態におけるプログラムローダ１は、プログラム全体を同時に配置できるだけの大

きなメモリを使用できるにも関わらず、あえてオーバーレイ技法を採用することで機密保護を実現する。すなわち、プログラムを複数の部分プログラムの集合の形で暗号化して保存しておき、プログラム実行にあたっては、この部分プログラムの単位で復号し、共有メモリＭの同一領域に順次上書きする形でロードする。さらに、部分プログラムの配置位置および配置タイミングの決定にあたって秘匿性を考慮するので、その分、機密保護能力が高い。

【００１３】

復号及びロードの処理単位である部分プログラムは、１つのソースファイル又は、互いに関連する複数のソースファイルのグループから生成されたオブジェクトコードを指す。オブジェクトコードとは、具体的には、プログラム構成部分を意味するサブプログラム、あるいはライブラリモジュールそのものを指す。

このように、部分に分けてロードすることで、不正参照が行われても、プログラム全体の内容が把握されてしまうおそれは小さくなる。全体を参照するには、各部分プログラムがロードされる度に不正参照を繰り返さなければならないからである。

【００１４】

ただし、本来はメモリ資源節約を目的とするオーバーレイには、機密保護に関する配慮がない。そのため、処理の終った部分プログラムは、次の部分プログラムが上書きされるまで、そのままメモリ上に残される。また、プログラムがロードされるエリアのアドレスが固定されているため、プログラムへのアクセスが容易に解読できてしまう。また、割込み機能を悪用したり、復号支援プログラムを改ざんしたりすれば、部分的とは言え、プログラムの不正な参照や改変も可能となる。

【００１５】

そこで、本実施の形態のプログラムローダ１では、オーバーレイの処理方式を、機密保護処理を考慮した形にアレンジして実行するほか、割込み機能の悪用や復号支援プログラムの改ざんチェックなどの処理も行う。

（構成）

（概要）

以下、プログラムローダ１の構成について説明する。

【００１６】

プログラムローダ１は、全体の処理を制御する制御部１１、復号支援プログラムＰを用いて暗号化部分プログラムなどの暗号化データの復号を行う復号部１２、復号に先立って復号支援プログラムＰの妥当性チェックを行う復号支援プログラム認証部１３、復号後の部分プログラムの共有メモリＭ上での配置位置および配置タイミングを決定するメモリ配置定義部１４、共有メモリＭ上で実行中の部分プログラムに対する割込みを用いた不正アクセスを防止する不正アクセス防止部１５を有する。

【００１７】

さらに、プログラムローダ１は、プログラム復号処理に用いられる各種の暗号化された情報、及び、暗号鍵を保持する保持部１６に加え、保持部１６が保持する格納アドレス情報（詳細は後述）の認証を行う格納アドレス情報認証部１７を有する。また、暗号化データ復号部１２は、扱うデータの種類に応じて、格納アドレス復号部１２１、メモリ配置情報復号部１２２、部分プログラム復号部１２３に分かれる。

【００１８】

保持部１６が保持する情報には、復号処理に使用される暗号鍵１６１、復号対象の暗号化部分プログラムの２次記憶装置Ｓにおける格納位置を示す情報である格納アドレス情報１６２、復号後の部分プログラムを共有メモリＭのどの位置に配置するかを示すメモリ配置情報１６３がある。

格納アドレス情報１６２は復号対象の暗号化部分プログラムを２次記憶装置Ｓから読み出す際に参照されるが、暗号化された形で保持されているので、格納アドレス復号部１２１によって復号される。さらに、復号後は格納アドレス情報認証部１７によって妥当性チェック（改ざん有無のチェック）が行われる。

【００１９】

図２は、格納アドレス情報１６２とそれが示す部分プログラムの２次記憶装置Ｓ上の格納位置との対応を示すイメージ図である。格納アドレス情報は、対応する部分プログラムの識別情報２１０と当該部分プログラムの２次記憶装置Ｓ上の

格納位置を示すアドレス情報２２０とからなり、部分プログラムの個数分存在する。このうち暗号化されているのはアドレス情報２２０の部分である。

【００２０】

また、メモリ配置情報１６３は、復号後の部分プログラムを共有メモリM上のどの位置に配置するかを規定する情報である。このように、あらかじめ配置位置を定めておくことは、データを所定の領域に順次上書きするオーバーレイ方式を実施するための前提である。メモリ配置情報１６３は暗号化された形で保持されており、メモリ配置情報復号部１２２によって参照の際に復号される。

【００２１】

図３はメモリ配置情報の構成と内容の一例とを示す概念図である。メモリ配置情報は、共有メモリM上に、どの部分プログラムをどのタイミング（順番）で配置するかを示す。共有メモリMの部分プログラム格納用領域は３つのエリアに分けられている。

図３に示すメモリ配置情報１６３は、エリア識別情報部３１０と部分プログラム識別情報部３２０とからなる。エリア識別情報部３１０には、いずれのエリアに関する情報かを示す識別情報が格納されており、部分プログラム識別情報部３２０には、エリア識別情報部３１０に格納された識別情報が示すエリアに配置される部分プログラムの識別情報が、配置される順番を示すデータとともに格納されている。

【００２２】

メモリ配置情報１６３は、共有メモリM内の部分プログラム格納用領域が３つのエリアに分割されていること、エリア１には３つ、エリア２には４つ、エリア３には２つの部分プログラムが順次配置されること、を示している。

図４は、プログラム実行時に、復号された部分プログラムが、メモリ配置情報１６３にしたがって、共有メモリM上にどのように配置されるかを示した概念図である。図４は、共有メモリM内に存在する３つの格納用エリアの各々に配置される部分プログラムを、プログラムの処理実行時間軸（横軸）に沿って示す。例えば、エリア１には、部分プログラムＡ，Ｂ，Ｃが順次配置され、時間帯ｔ１においては、部分プログラムＡ（エリア１）、部分プログラムD（エリア２）、部

分プログラムＨ（エリア３）が共有メモリＭ上に共存している。同じ時間帯に共有メモリＭに共存する部分プログラムには、それぞれが処理の過程で他の部分プログラムの処理を呼び出すなどの形で依存関係を有するものである。

【００２３】

こうしたメモリ配置情報は、メモリ配置定義部１４が、暗号化対象プログラムの暗号化処理に先立ってあらかじめ生成しておくものである。生成処理は従来のオーバーレイ方式において使用される同種の配置情報と基本的に同じであるが、本実施の形態では機密保護も考慮したやり方で生成される。本実施の形態における生成処理の特徴については後述する。

【００２４】

（主要構成部の説明）

以下、上で概要を述べた構成部のうち主要なものについて、さらに詳しく説明する。

＜復号支援プログラム認証部１３＞

復号支援プログラム認証部１３は、各種暗号化データの復号に先立って、復号支援プログラムＰの正当性を認証する。具体的には、復号支援プログラム認証部１３は、部分プログラム復号部１２３が暗号化部分プログラムを復号しようとする場合、格納アドレス復号部１２１が暗号化状態の格納アドレス情報を復号しようとする場合、メモリ配置情報復号部１２２が暗号化状態のメモリ配置情報を復号しようとする場合に、これら構成部からの要求に応じて復号支援プログラムＰの認証処理を行う。復号支援プログラム認証部１３は認証処理の結果を要求元に返し、認証結果が「正当（改ざん無し）」であった場合は、要求元から復号対象データを受け取って復号支援プログラムＰに送り、さらには、復号処理結果を復号支援プログラムＰから受け取って要求元に送る。

【００２５】

復号支援プログラム認証部１３による認証処理は、復号支援プログラムＰが以前の実行時から現在までの間に改ざんされていないかを確認するもので、判定の基準として、復号支援プログラムＰのサイズ変化の有無、更新日時、あるいは復号支援プログラムＰの一方向ハッシュ値などを参照する。ただし、手法はこれら

に限定されず、電子署名認証技術などの技術を用いてもよい。復号支援プログラム認証部１３は、この認証処理のために、復号支援プログラムＰの初回実行時のサイズや更新日時情報、ハッシュ値を保持しておく。

【００２６】

＜格納アドレス情報認証部１７＞

格納アドレス情報認証部１７は、格納アドレス復号部１２１からの指示に応じて復号支援プログラムＰが復号した格納アドレス情報の正当性を認証する。これは、以前の実行時から現在までの間に格納アドレス情報が改ざんされていないかを確認するものである。

【００２７】

格納アドレス情報認証部１７は、一方向ハッシュ関数等、一般に用いられている認証技術を用いて格納アドレス情報の認証処理を行い、結果を格納アドレス復号部１２１に返す。格納アドレス情報認証部１７は、この認証処理のために、各格納アドレス情報について必要な情報（初回復号時のハッシュ値、認証処理に用いる情報全般）を保持しておく。

【００２８】

＜制御部１１＞

制御部１１は、復号すべき部分プログラムをＯＳから指示されると、対象の部分プログラムの識別情報を格納アドレス復号部１２１に送って、当該部分プログラムの格納アドレスを取得、復号するように指示する。そして、格納アドレス復号部１２１から復号後の格納アドレスが出力されてくると、これを部分プログラム復号部１２３に渡し、当該部分プログラムを復号するよう指示する。また、それと平行して、メモリ配置情報復号部１２２に処理対象の部分プログラムの識別情報を送り、当該部分プログラム用のメモリ配置情報を復号するよう指示する。

【００２９】

そして、制御部１１は、部分プログラム復号部１２３から出力されてきた復号後の部分プログラムを、メモリ配置情報復号部１２２から出力されてくるメモリ配置情報に従って、共有メモリＭ上のエリアのいずれかにロードする。なお、メモリ配置情報は、図３に示すとおり、部分プログラムの識別情報とエリアの識別

情報とからなっていて、各エリアの絶対アドレス値を示す情報は含まれていない。各エリアの絶対アドレス値については、制御部１１が保持しておくものとする。

【００３０】

また、制御部１１は、上記処理の過程で復号支援プログラムＰや格納アドレス情報に問題が生じた場合（正当性が認証できなかった場合）、他構成部に対して処理停止を指示するとともに、その時点で復号済みの各種データ（部分プログラム、メモリ配置情報、格納アドレス情報）を消去するための処理を行う。

＜メモリ配置定義部１４＞

メモリ配置定義部１４は、復号処理において上記のように参照されるメモリ配置情報１６３を、部分プログラムの暗号化処理時に生成して、保持部１６に格納しておく。部分プログラムの暗号化は、暗号化プログラム生成装置Ｃが行う。暗号化プログラム生成装置Ｃは、メモリ配置情報に設定された順序で各部分プログラムがメモリに配置されるように暗号化対象のプログラム（部分プログラムの集合）を実行形式に変換したうえで暗号化し、２次記憶装置Ｓに格納する。

【００３１】

まず、メモリ配置定義部１４は、ヘッダ情報などから、部分プログラムのメモリ配置の決定に必要な各種情報を得る。さらには、部分プログラムの実行時の配置先である共有メモリＭに関する情報（部分プログラム格納領域及びそれに含まれるエリアの数及びサイズ）など、決定時に重視すべき条件（パラメータとしてシステム管理者から指定されるもの）を得て、これら情報からメモリ配置情報を生成する。メモリ配置情報生成時にメモリ配置定義部１４が参照する情報は、具体的には下記のようになる。

（１）各部分プログラムのサイズ（暗号化前、すなわち復号後のサイズ）

（２）部分プログラム間の依存関係（コールする側とコールされる側との関係、及びコール回数）

（３）各部分プログラムの秘匿性のレベル

（４）求められるパフォーマンスのレベル

このような情報を元にメモリ配置を決定する処理は、従来のオーバーレイ処理

実行においても行われるものであるが、本実施の形態では、プログラムの機密保護を目的としているので、通常のオーバーレイを想定したメモリ配置の場合に比べて、特に（３）を重視し、（４）の優先度を低くする。

【００３２】

そこで、メモリ配置定義部１４は、秘匿性の高い部分プログラムについては、こまめにロードと消去とを繰り返して、必要最小限の時間だけ共有メモリＭ上に配置し、処理が終れば直ちに別の部分プログラムが上書きされる、という形になるようなメモリ配置情報を生成する。部分プログラムの秘匿性のレベルについては、システム管理者が部分プログラムごとに評価し、その秘匿性の評価値をパラメータとしてメモリ配置定義部１４に入力する、などの設定の仕方が考えられる。

【００３３】

さらに言えば、以下のようなメモリ配置情報設定方法も考えられる。すなわち、メモリ配置定義部１４は、従来のメモリ配置情報決定のアルゴリズムを用いて複数パターンの候補配置情報を生成する。そして、それら複数の候補配置情報を機密保護の観点から定めた基準に従って評価し、もっとも優れたものを配置情報とする。基準としては、「秘匿性の高い部分プログラムがメモリ上に存在する時間長の予測値」が考えられる。または、メモリ配置定義部１４が生成した候補配置情報をシステム管理者が参照して、その中から１つを選択してもよい。

【００３４】

また、同時にメモリ上に展開されている部分プログラムのサイズの合計値（一度に不正参照できるプログラムの大きさ）を小さくしたい場合は、部分プログラムがロードされるメモリのサイズを小さく設定すればよい。

なお、メモリ配置定義部１４は、プログラムローダ１ではなく暗号化プログラム生成装置Ｃの構成要素とすることもできる。

【００３５】

＜不正アクセス防止部１５＞

不正アクセス防止部１５は、共有メモリＭ上で実行されている部分プログラムに対する、割り込みを利用した、不正なプログラム解析を防止するための処理を

行う。

割り込みとは、コンピュータシステム上である処理の実行中に、より優先度の高い他イベントが発生することである。一般的に、割り込みが発生すると、実行中の処理は一時的に中断され、割り込みイベントに対応した処理が行われることになる。この機能を利用すれば、プログラムを任意の箇所で停止させ、その時点でのメモリやレジスタの内容を参照したり、あるいは変更したうえで処理を再開させたりすることが可能になる。

【００３６】

例えば、本プログラムローダ１が格納アドレス情報の復号を行った直後に割り込みを発生させれば、復号された格納アドレス情報を参照することができる。また、部分プログラムの復号及びロードが完了した後で割り込みを発生させれば、共有メモリＭ上に配置された部分プログラムの内容が参照可能になる。

不正アクセス防止部１５は、こうした事態を防止するために、割り込みが検知されると、他の構成部が実行中の処理を中止させたり、共有メモリＭ上にロードされている部分プログラムを消去させたりする。

【００３７】

具体的には、不正アクセス防止部１５は、割り込みを検知すると、プログラムローダ１が存在するコンピュータシステム上で実行中のプログラムの処理を中断させるトラップ命令を発行し、ＣＰＵのＩＤＴ（Interrupt Descriptor Table）を参照する。ＩＤＴテーブルには割り込み命令に対応したハンドラの情報が定義されている。本実施例では、プログラムの中止、共有メモリＭの復号部分プログラム格納領域にある内容の消去といった動作を実行するためのハンドラが定義されている。そして、不正アクセス防止部１５は、上記ハンドラに処理を移し、プログラムの中断、メモリ内容の消去を実行させる。

【００３８】

また、不正アクセス防止部１５は、上記の処理と平行して、ダミープログラム実行のための処理を行う。ダミープログラムは、上記のプログラム中断及びメモリ内容の消去の間、不正割り込みを行って制御を奪おうとする者の注意をそらすためのものである。ダミープログラムの処理内容は、他プログラムの処理に影響

を与えないもの、例えば、「文字列の表示のみ行う」又は「暗号化プログラムの実行結果と反対の結果を出力する」といった内容としておく。不正アクセス防止部１５は、割り込みを検知すると、予め実行環境にロードされているダミープログラムをコールすることで、これを実行させる。

（動作）

次いで、上記の構成を有するプログラムローダ１の動作について、図面を参照しながら、制御部１１を中心に説明する。

【００３９】

図５は、暗号化プログラムの復号及びロード処理におけるプログラムローダ１の動作を示すフローチャートである。ただし不正アクセス防止部１５の処理については、割り込み的に実行されるものなので本図には示していない。また、プログラムの暗号化、それに平行して行われるメモリ配置情報の生成処理も本図の対象外である。

【００４０】

復号・ロード処理は、制御部１１が、外部（ＯＳ又はシステム上で実行中のプログラム）から処理実行指示（処理対象の部分プログラムを指定する識別情報を含むもの）を受け取った時点で開始される（Ｓ５０１：Yes）。ＯＳから処理実行指示が送られてくるのは、対象プログラムの起動の際、そして、起動後、復号・ロードされて実行中の部分プログラムが他の部分プログラムを呼び出した場合である。起動時の指示では、プログラムのエントリポイントを持つ部分プログラムが復号の対象となり、起動後の指示では、呼び出された部分プログラムの識別情報が処理対象となる。

【００４１】

指示を受けた制御部１１は、指定された部分プログラムを２次記憶装置Ｓから読み出させるために、格納アドレス復号部１２１に対して指定された部分プログラムの識別情報を渡し、格納アドレス情報の復号を指示する。

格納アドレス復号部１２１は、制御部１１からの指示を受けると、保持部１６から当該部分プログラム用の格納アドレス情報（暗号化状態）を読み出すと共に（Ｓ５０２）、復号支援プログラム認証部１３に指示して復号支援プログラムＰ

の正当性認証処理を行わせる。

【００４２】

復号支援プログラムＰの正当性が認証された場合（Ｓ５０３：Yes）、格納アドレス復号部１２１は、ステップＳ５０２で読み出した格納アドレス情報を、復号支援プログラムＰを用いて復号する。その際、格納アドレス復号部１２１は、保持部１６から当該情報復号用の暗号鍵を取得して、これを暗号化格納アドレス情報と共に復号支援プログラムＰに送り、復号処理を行わせる（Ｓ５０４）。

【００４３】

逆に、復号支援プログラムの正当性が認証されなかった（改ざんありと判定された）場合（Ｓ５０３：No）、復号支援プログラム認証部１３はその旨を制御部１１に通知し、制御部１１はＯＳに改ざんの検出を通知した上で、復号・ロード処理を中断する（Ｓ５１４）。

ステップＳ５０４で復号後の格納アドレス情報を得た格納アドレス復号部１２１は、これを格納アドレス情報認証部１７に送り、正当性の認証処理を行わせる。格納アドレス情報認証部１７は、一方向ハッシュ関数等、一般に用いられている認証技術を用いて認証処理を行い、結果を格納アドレス復号部１２１に返す。格納アドレス復号部１２１は結果を制御部１１に送る。

【００４４】

認証処理の結果、格納アドレス情報の正当性が確認されなかった場合（Ｓ５０５：No）、制御部１１はＯＳに「格納アドレスに不正がある」旨を通知して、復号・ロード処理を中止する（Ｓ５１４）。

一方、格納アドレス情報の正当性が認証された場合（Ｓ５０５：Yes）、制御部１１は、復号された格納アドレス情報を格納アドレス復号部１２１から得て、復号・ロード対象の部分プログラムを２次記憶装置Ｓから読み出して復号するための処理に移る。具体的には、先ず、復号処理を実行する部分プログラム復号部１２３に対し、処理開始を指示する。

【００４５】

部分プログラム復号部１２３は、制御部１１からの指示を受けると、先ず、指示された暗号化部分プログラムと部分プログラム復号用の暗号鍵とを２次記憶装

置Ｓの格納位置から読み出す（Ｓ５０６）。そして、復号支援プログラム認証部１３に指示して復号支援プログラムＰの正当性認証処理を行わせ、その結果を制御部１１に返す。

【００４６】

復号支援プログラムＰの正当性が認証された場合（Ｓ５０７：Yes）、部分プログラム復号部１２３は、ステップＳ５０６で得た暗号化部分プログラムを暗号鍵と共に復号支援プログラムＰに送って当該部分プログラムの復号処理を行わせる（Ｓ５０８）。

逆に、復号支援プログラムＰの正当性が認証されなかった場合（Ｓ５０７：No）、その結果は部分プログラム復号部１２３から制御部１１へ伝えられ、制御部１１は、ＯＳに「復号支援プログラムの改ざん検出」を通知し、当該部分プログラムの復号・ロード処理を中断する。また、この時点で共有メモリM上にロード済みの復号部分プログラムがあれば、それを消去し、さらに、復号された格納アドレスも消去させる（Ｓ５１４）。

【００４７】

部分プログラムの復号処理を終えると、部分プログラム復号部１２３は、復号した部分プログラムを制御部１１に出力する。制御部１１は、これを共有メモリ空間M上に配置するための処理を行う。具体的には、先ず、制御部１１はメモリ配置情報復号部１２２に指示して、保持部１６が保持する暗号化メモリ配置情報を暗号鍵と共に読み出させる（Ｓ５０９）。

【００４８】

メモリ配置情報復号部１２２は、復号支援プログラムＰの正当性認証処理を復号支援プログラム認証部１３に行わせる。

復号支援プログラムＰの正当性が認証された場合（Ｓ５１０：Yes）、メモリ配置情報復号部１２２は、暗号化状態のメモリ配置情報を暗号鍵とともに復号支援プログラムＰに送って復号処理を行わせ、復号後のメモリ配置情報を制御部１１に返す（Ｓ５１１）。制御部１１は、復号されたメモリ配置位置情報に基づいて、ステップＳ５０８で得た復号後の部分プログラムを共有メモリM上に配置する（Ｓ５１２）。

【００４９】

ステップＳ５１０で復号支援プログラムＰの正当性が認証されなかった場合（Ｓ５１０：No）、メモリ配置情報復号部１２２は、その旨を制御部１１に通知する。制御部１１は、ステップＳ５０８で復号された部分プログラムを消去して、処理を中断する（Ｓ５１４）。なお、この時点でメモリ空間上にロード済みの部分プログラムがあれば、それも消去する。

【００５０】

以上の処理は、対象プログラム全体の処理が終了するまで繰り返される（Ｓ５１３：Yes）。

（まとめ）

上記の通り、本実施の形態におけるプログラムローダ１は、復号対象の暗号化プログラムに関して、部分プログラムの単位で分けて復号を行い、復号した部分プログラムは、機密保護を考慮して設定されたメモリ配置情報に従ってメモリ上の所定エリアに上書き配置する、というオーバーレイ方式を採用することで、プログラム全体が不正参照されることを防止している。さらに、以下のような処理を行うことで、機密保護の厳密化を図っている。（１）個々の部分プログラム復号のたびに復号支援プログラムの正当性をチェックすることで、復号支援プログラムの改ざんによる不正をチェックする。（２）復号後の部分プログラムのメモリ配置を示すメモリ配置情報も暗号化しておくことで、復号後のプログラムの不正参照を困難にする。（３）割込み検知時は復号済みのデータ（部分プログラムなど）を消去することで、割込みを用いた不正参照を防止する。

（変形例）

以下、上記の実施の形態に関して考えられる５つの変形例について説明する。

【００５１】

（変形例１）

上記の実施の形態では、不正アクセス防止部１５が、割り込み発生を常に不正なものと判断して、処理停止と復号済みデータの消去とが行われているが、割り込みの中には正当なものもある。そこで、本変形例では、正当な割り込みについては認めるよう処理を行う。

【００５２】

本変形例における暗号化データ復号装置全体の構成については、上記実施の形態と同じであり、不正アクセス防止部の処理内容が部分的に異なるのみである。よって構成図は省略するが、区別のために、以下、「不正アクセス防止部１５’」と言う。

本変形例における不正アクセス防止部１５’は、正当なデバッガのための割り込みを認めることで、割り込みの不正使用によるプログラムの逆解析を防止する一方で、正当な用途である不良箇所調査などのためのデバッガに関連する割り込みについては可能とする。デバッガは不良調査のために、プログラムの処理を所望の位置で停止させて（ブレークして）、その状態でのメモリの内容を参照したり書き換えたりできるようにする機能である。

【００５３】

ブレークによるプログラム停止は、ブレークを発生させるブレークポイントを予め設定しておき、処理がブレークポイントに到達するたびに、割り込みが発生させられてプログラムが停止させられる、という形で行われる。またブレークポイントにおいてプログラムを停止させる前提となる条件を詳細に設定することもできる。しかし、上記実施の形態における不正アクセス防止部１５の処理方式では、ＯＳがブレークのための割り込みを発生させた時点でプログラムが停止されるので、不良調査はできない。

【００５４】

そこで、本変形例での不正アクセス防止部１５’は、デバッガに関わる割り込み（ブレーク用割り込み）とそれ以外の割り込みとを下記のような手順で区別する。

不正アクセス防止部１５’は、暗号化対象のプログラムが暗号化プログラム生成装置Ｃによって実行形式に変換された後、これが暗号化される前の段階で、正当な権限を有する利用者から、実行形式プログラムに対するブレークポイントの設定を受け付ける。そして、設定したブレークポイントの位置情報（行番号、関数名、アドレス等で表されるもの）を保持しておく。

【００５５】

そして、プログラムの復号及びロードの段階において、不正アクセス防止部１５’は割り込みを監視する（不正アクセス防止部１５と同じ）。そして、プログラムの実行途中で割り込みが発生すると、発生位置を、保持しているブレークポイント位置情報に照会する。そして、発生位置がブレークポイント位置情報で予め設定されていたブレークポイントの位置に一致していれば、実施の形態で述べたようなメモリ内容の消去などの処理は行わず、利用者に実行を続けさせる。

【００５６】

逆に、予め設定されていた位置と異なる位置で割り込みが発生していた場合、不正アクセス防止部１５’は、不正アクセス防止部１５と同様に、プログラムの実行を中断して、メモリＭの部分プログラムなどの復号済みデータを消去するための処理を行う。

（変形例２）

次に、メモリ配置情報の機密保護をさらに厳密に行う変形例について説明する。

【００５７】

上記の実施の形態では、いったん生成されたメモリ配置情報は、対象となる暗号化プログラムの復号及びロードが行われる度に毎回同じ内容の情報が使用される。つまり、ある暗号化プログラムの実行時、これを構成する部分プログラムの各々が、毎回、同じタイミングでメモリ空間上の同じアドレスに配置される。そのため、メモリ空間上のデータを監視しながらプログラム繰り返し実行することで、メモリ配置情報の内容が解読されてしまう可能性も皆無とはいえない。メモリ配置情報が解読されると、これを利用して不正参照が行われることになる。

【００５８】

そこで、本変形例では、メモリ配置情報が解読されることを防止し、それによって、暗号化プログラムの機密保護をより確実に行えるようにする。そのために、本変形例では、プログラムの実行のたびに、これを構成する部分プログラムのメモリ配置位置が動的に変化するように、メモリ配置情報を生成する。

具体的には、共有メモリＭ内の３つのエリアそれぞれに割り当てる絶対アドレスの値が、プログラム実行のたびに切り替わるようにメモリ配置情報を設定する

【００５９】

本変形例における暗号化データ復号装置全体の構成については、上記実施の形態と同じであり、メモリ配置定義部の処理内容が部分的に異なるのみである。よって装置全体の構成図は省略するが、区別のために、以下、本変形例におけるメモリ配置定義部については、「メモリ配置定義部１４’」とする。

図６は、メモリ配置定義部１４’がエリアのアドレス切り替えのために保持する３パターンのエリアアドレス情報６０１、６０２、６０３を示す。各エリアアドレス情報は、エリアとこれに割り当てられる絶対アドレスとの組み合わせで成り、エリア識別子欄６１０と絶対アドレス値欄６２０とから成る。

【００６０】

エリアアドレス情報に相当する情報は、上記実施の形態では制御部１１が１種類だけ（例えば、エリアアドレス情報６０１）保持して使用していた。本変形例では、メモリ配置定義部１４’が３パターンのエリアアドレス情報から１つを選び、メモリ配置情報生成時に、制御部１１に渡しておく。制御部１１は、部分プログラムの復号及びロード時、選ばれたパターンのエリアアドレス情報の内容に従って各エリアの絶対アドレスを決定し、さらには、各エリアの絶対アドレスとメモリ配置情報とから、復号後の部分プログラムの共有メモリＭにおける配置アドレスを得る。

【００６１】

メモリ配置定義部１４’が、エリアアドレス情報のパターン選択を行うタイミングは、プログラムの実行指示がプログラムローダ１に入力されたタイミングでもよいし、プログラムの復号及びロード処理が開始されて、最初の部分プログラムのロードを行うタイミングでもよい。

なお、エリアの絶対アドレスの切り替えは、必ずしも、上記のようなパターン切り替えという形で行う必要はない。絶対アドレスを「ずらす」形でメモリ配置位置を変えることもできる。

【００６２】

図７は、「ずらす」形での切り替えに使用されるエリアアドレス情報を示す。

エリアアドレス情報７００は、各エリアと絶対アドレス値とを１対１で対応づけているが、各絶対アドレス値が変数（ずらし幅）α、β、γを含んでおり、メモリ配置定義部１４’は、これら変数の値を変更することで、各エリアの絶対アドレスを切り替える。

【００６３】

変数の値は乱数発生プログラム（乱数発生関数、タイマの保持する時刻情報を用いるもの、など）で任意の値に決定すればよいが、「あるエリアに対応する変数（ずらし幅）の値と当該エリアに配置される部分プログラムのサイズとの和は、当該エリアのサイズ以下である」という条件に従って決定する。これは、ずらすことによって、部分プログラムがエリア外にはみ出す形で配置される事態を防止するためである。

【００６４】

なお、メモリ配置定義部１４’は、直前のプログラム実行時のエリアアドレスの内容を記憶しておき、それを参照することで、各エリアの絶対アドレスが必ず直前の実行時のアドレスと異なるようにすること、としてもよい。

なお、こうしたエリアアドレスの変更は、上記のようにメモリ配置定義部１４’が保持する情報に従って制御部１１が行うのでなく、制御部が単独で行うこともできる。

【００６５】

（変形例３）

次に、特定の部分プログラムが長時間共有メモリM上に配置されたままになることを防止することで、メモリ上での部分プログラムの機密保護をさらに厳密に行う、という変形例について説明する。

実施の形態の場合のように、同一エリアに部分プログラムを上書きするオーバーレイを採用しているのは、特定の部分プログラムが長時間共有メモリ上に配置されたままになっていると、その分、不正参照される危険性が高まるためである。しかし、上書きされる部分プログラムのサイズによっては、オーバーレイ方式を採用していても、処理の終った部分プログラムの一部が消去されずにメモリ上に残ってしまう。

【００６６】

例えば、図４に示す形で部分プログラムのメモリ配置が行われた場合、エリア１に順次配置された３つの部分プログラムのうち、最初に配置された部分プログラムＡのサイズに比べ、後から上書きされた部分プログラムＢ、Ｃそれぞれのサイズがその半分しかなかったとする。すると、部分プログラムＡの後半部分の内容は、部分プログラムＢ、Ｃの上書きによっては消去されず、エリア１に残ることとなる。もし部分プログラムＡの後半が特に秘匿性の高い内容であれば、このように長時間配置されたままになることは機密保護の観点から好ましくない。

【００６７】

また、エリア３については、時間帯ｔ１で部分プログラムＨの処理は完了しているにも関わらず、時間帯ｔ２の開始時点から次の部分プログラムＩが配置されるまでの間、部分プログラムＨのデータが残されてしまう。

本変形例では、１つの部分プログラムを復号してメモリ空間に配置するたびに、配置先のエリアのサイズと当該部分プログラムのサイズとの差分の有無をチェックし、エリアサイズの方が大きい場合は、エリアの空き領域部分にダミーデータ（ダミープログラム）を埋め込む（上書きする）ことで、上記の問題の解消を図る。また、ある部分プログラムの処理が完了すると、そこから経過時間を計測し、経過時間が長くなった場合には、当該エリア全体にダミーデータを上書きする。上記処理は制御部が行う（以下、実施の形態における制御部１１と区別して「制御部１１’」とする）。

【００６８】

図８は、こうしたダミーデータ埋め込み処理の概要を示すイメージ図であり、特定のエリアＮにおける部分プログラムの配置、及びダミーデータの埋め込みの状況を時間経過に沿って示している。時間帯Ｔ２、Ｔ３、Ｔ５においては、それぞれで配置された部分プログラムのサイズが小さいので、エリアサイズと部分プログラムサイズとの差分にダミーデータが埋め込まれている。また、時間帯Ｔ４では、直前の部分プログラム終了後に生じる空き時間が長いため、エリア全体にダミーデータが埋め込まれている。

【００６９】

制御部１１’は、部分プログラム復号部１２３が復号した部分プログラムを、メモリ配置情報に従って共有メモリM内のいずれかのエリアに配置する時点で、当該部分プログラムの復号後のサイズ（メモリ配置定義部１４’がメモリ配置情報の一部として設定しておくもの）を得るとともに、配置先エリアのサイズを、共有メモリMを参照して得る。

【００７０】

制御部１１’は、これら２つのサイズの値を比較し、エリアのサイズの方が大きければ、両サイズの差分に応じた量だけ、ダミープログラムデータをダミープログラム格納領域から読み出し、当該エリアの空き領域（部分プログラムの終端以降の領域、及び／又は、先頭より前の領域）に埋め込む。これによって、同エリアに直前に配置されていた部分プログラムのデータは完全に消去される。

【００７１】

空き時間に対応してのダミーデータ埋め込みは、ある部分プログラムの処理が完了した後に行われる。制御部１１’は、あるエリアにロードされていた部分プログラムの処理が終了すると、当該エリア用の内蔵タイマで経過時間の計測を開始し、経過時間が所定値に達した時点で、次に当該エリアにロードされる部分プログラムの復号が行われていなければ、当該エリア全体にダミーデータを埋め込む。

【００７２】

これによって、同エリアに直前に配置されていた部分プログラムのデータは完全に消去される。ダミーデータの実体は、実際には実行されないプログラム、又は、実行されても意味のある処理は行わないプログラムのコードとする。

（変形例４）

本変形例においては、暗号鍵を格納する場所、及び取得のための手順について工夫することで、部分プログラムごとに機密保護を強化する。具体的には、各部分プログラムの復号用の暗号鍵を、当該部分プログラムに先立って共有メモリMに正当にロードされている別の部分プログラムから取得するようにする。

【００７３】

本変形例では、各部分プログラムの中に、当該部分プログラムがコールする部

分プログラムの復号用の暗号鍵そのもの、又は暗号鍵を得るためのデータを埋め込んでおき、新たにコールされた部分プログラムの復号時には、コール元部分プログラムに埋め込まれた暗号鍵又は暗号鍵を得るためのデータを使用する。

＜基本形＞

この方式の実施様態として最も基本的なものは、以下の通りである。先ず、暗号化プログラム生成装置Ｃが部分プログラムを暗号化する前に、当該部分プログラムによってコールされる別の部分プログラムの暗号化に用いる暗号鍵を埋め込んだ上で暗号化し、さらに、その暗号鍵の埋め込み位置の情報を保持部１６に格納しておく。

【００７４】

そして、部分プログラム復号の際は、部分プログラム復号部１２３が、復号対象部分プログラムのコール元である部分プログラムに関する埋め込み位置情報を保持部１６から読み出し、その埋め込み位置情報を元に、共有メモリＭ上にある当該コール元部分プログラムから暗号鍵を読み出し、これを使って復号対象の部分プログラムを復号する。

【００７５】

例えば、不正利用者が制御を不正に奪って所望の部分プログラムをメモリ上にロードさせるためのコール命令を発行させたとする。その場合、正当なコール元の部分プログラムは共有メモリＭ上に存在しないため、本変形例の方式では暗号鍵が得られず、復号もできない。よって、不正利用者が望んだ部分プログラムの共有メモリＭへのロードは行われず、不正参照は防止される。

＜応用例＞

本変形例の基本的な様態は上記の通りであるが、さらに手を加えて機密保護の厳密化を図ることもできる。そうした応用例について以下に説明する。

【００７６】

＜応用１＞

図９は、暗号化処理を行う構成部（暗号化プログラム生成装置Ｃ９）と復号処理を行う構成部（部分プログラム復号部９２３）とで同じ暗号鍵生成手段（鍵生成部９０１ａ、９０１ｂ）を共有し、コール元の部分プログラムのコードの一部

から、コール対象の部分プログラムの暗号化／復号に用いられる暗号鍵を生成する、という本例のプログラムローダ９を示す。なお、実施の形態におけるプログラムローダ１と同一の処理を行う構成については同一の参照番号を付して説明は省略する。

【００７７】

・暗号化時の処理

暗号化プログラム生成装置Ｃ９は、まず、各部分プログラム間の呼出関係を示す呼出関係情報を生成し、さらに各部分プログラムを実行形式に変換する。そして、呼出関係情報を参照しながら各部分プログラム用の暗号鍵を生成する。そして、この暗号鍵を用いて部分プログラムの暗号化を行う。

【００７８】

図１０は、部分プログラム間の呼出関係を示す模式図である。

暗号化プログラム生成装置Ｃ９が、１つの部分プログラムを暗号化する処理を以下に説明する。先ず、前記呼出関係情報を参照して、当該部分プログラムを呼び出すコール元部分プログラムを検出する。そして、コール元部分プログラムの実行形式コードの一部を読み出す。そして、読み出したコードを鍵生成部９０１ａに渡し、暗号鍵生成を指示する。鍵生成部９０１ａは、当該コードのハッシュ値を算出し、これを暗号鍵として暗号化プログラム生成装置Ｃ９に返す。暗号化プログラム生成部Ｃ９は、この暗号鍵を用いて部分プログラムを暗号化すると、暗号化した部分プログラムに、暗号鍵生成に用いたコードの位置（当該コードのコール元部分プログラムの実行形式におけるオフセット）を示す情報を付加する。そして、２次記憶装置Ｓに、この暗号化した部分プログラムを格納する。

【００７９】

なお、１つの部分プログラムが複数の部分プログラムから呼び出される場合、当該部分プログラムは、それぞれのコール元部分プログラムのコードから得られる暗号鍵によって暗号化される。よって、１つの部分プログラムから複数パターン暗号化部分プログラムが生成されることになる。その場合は、それぞれの暗号化部分プログラムに対応するコール元部分プログラムの識別情報を付加した上で２次記憶装置Ｓに格納するなどして、復号処理を行う構成部が対応関係を把握で

きるようにしておく必要がある。ただし、複数の部分プログラムからコールされる場合でも、暗号鍵が必要なのは対象部分プログラムがメモリ上にない状況でコールされる場合のみであるから、対象部分プログラムが必ずメモリ上にある状況でこれをコールするコール元部分プログラムからは暗号鍵を生成する必要がない。

【００８０】

例えば、図１０に示す部分プログラムＨの場合、コール元である部分プログラムＣ、Ｆから暗号鍵が生成される。ただし、（１）「部分プログラムＦからコールされることで部分プログラムＨは復号・ロードされる」、しかも、（２）「部分プログラムＣが部分プログラムＨをコールする時には、すでに部分プログラムＨはメモリにロード済みであることが、メモリ配置情報により規定されている」のであれば、部分プログラムＨ復号用の暗号鍵は部分プログラムＦからのみ生成すればよく、部分プログラムＣから暗号鍵を生成する必要はない。

【００８１】

・復号時の処理

次いで、プログラムの実行段階において、部分プログラム復号部９２３が、１つの暗号化部分プログラムを復号する処理を以下に説明する。まず、部分プログラム復号部９２３は、復号して共有メモリＭ上にロードすべき部分プログラムの識別情報を、そのコール元の部分プログラムの識別情報とともに制御部９１１から受け取る。そして、まず、２次記憶装置Ｓから復号対象の暗号化部分プログラムを読み出す。

【００８２】

そして、部分プログラム復号部９２３は、この暗号化部分プログラムから、暗号鍵の元になるコードの位置を示す情報を読み出すと、共有メモリＭにロードされているコール元部分プログラムから、当該コード位置が示す位置のコードを読み出す。その後、鍵生成部９０１ｂに読み出したコードを渡して暗号鍵を生成させ、この暗号鍵を用いて対象の暗号化部分プログラムを復号する。復号を終えると、暗号鍵を直ちに消去する。

【００８３】

図１１は、復号処理時の部分プログラムのメモリ配置を時間軸に沿って示す模式図である。この図に示す例では、部分プログラムＨは、先にエリア２にロードされていた部分プログラムＦからコールされることでエリア３にロードされる。そしてその後、エリア３にロード済みの部分プログラムＨを、後からエリア１にロードされた部分プログラムＣがコールする、という順序になっている。

【００８４】

よって、部分プログラム復号部９２３は、部分プログラムＦから生成された暗号鍵を用いて部分プログラムＨを復号し、部分プログラムＣが部分プログラムＨをコールした時点では、復号処理は行わない。部分プログラムＣによるコール時に復号処理が不要と判定するのは、制御部９１１でも部分プログラム復号部９２３でもよい。判定は、コール先の部分プログラムの識別情報をメモリ配置情報に照会することで行ってもよいし、コール先部分プログラムがメモリＭ上にロード済みか否かをチェックすることで行ってもよい。

【００８５】

＜応用２＞

次に、もう１つの応用例について説明する。

図１２示す本応用例の暗号化プログラム生成装置Ｃ１２は、暗号化処理において、コール元部分プログラム内に、コール対象部分プログラム用の暗号鍵を取得する鍵取得プログラムを埋め込んでおく（鍵取得プログラム生成部１２０１、暗号化プログラム生成装置Ｃ１２）。そして、復号時、部分プログラム復号部１２２３は、復号対象部分プログラムのコール元部分プログラムに埋め込まれた鍵取得プログラムを実行することで暗号鍵を取得し、これを使用して当該復号対象部分プログラムの復号を行う。

【００８６】

・暗号化時の処理

暗号化プログラム生成装置Ｃ１２が、暗号化対象のプログラムを実行形式に変換した上で部分プログラムに分割するまでの処理は、既に述べた実施の形態の場合と同じである。以下、暗号化プログラム生成装置Ｃ１２が、１つの部分プログラムの暗号化にともなって行う処理を説明する。

【００８７】

暗号化プログラム生成装置Ｃ１２は、暗号化対象の部分プログラムに関して暗号鍵Kを生成する。そして、暗号鍵を予め定めた格納位置に格納し、格納位置を示す情報（アドレス、オフセット等）を保存する。この格納位置は、２次記憶装置Ｓ内の所定領域、あるいは、当該部分プログラム復号時に共有メモリＭ上にロードされている他の部分プログラムの中などである。そして、当該格納位置を鍵取得プログラム生成部１２０１に渡して、「この格納位置から暗号鍵Kを読み出して、保持部１６内の所定位置（当該部分プログラム用の暗号鍵K格納に割り当てられた領域）に書き出す処理」を行う鍵取得プログラムを生成させる。

【００８８】

ある部分プログラムに関して上記の処理を完了した暗号化プログラム生成装置Ｃ１２は、当該部分プログラムのコール元となる別の部分プログラムに鍵取得プログラムを埋め込む。埋め込み位置は、例えば、当該部分プログラムがコールされる処理の直前とする。

全部分プログラムについて、鍵取得プログラムの生成とコール元部分プログラムへの埋め込みとを終えると、暗号化プログラム生成装置Ｃ１２は、部分プログラムの暗号化を行う。

【００８９】

・復号時の処理

復号時の暗号鍵取得のための処理は、実施の形態の場合とほとんど同じである。部分プログラム復号部１２２３は、ある部分プログラムの復号指示を受けると、格納部１６内の領域で、当該部分プログラム用暗号鍵Kの格納用に割り当てられた領域を参照して暗号鍵Kを読み出し、これを用いて復号を行う。当該暗号鍵Kは、既に共有メモリＭ上にロード済みのコール元部分プログラムに埋め込まれた鍵取得プログラムが実行されたことによって、当該領域に書き出されていたものである。なお、復号を終えた部分プログラム復号部１２２３は、当該領域から暗号鍵Kを消去しておく。同じ部分プログラムがまたコールされることがあるとしても、その時には、コール元に埋め込まれた鍵取得プログラムが同じ領域に書き出してくれるので消去しても問題はない。

【００９０】

以上のようにすることで、本変形例では、保持部など所定の場所に全部分プログラム共通の暗号鍵を恒常的に保管しておく場合に比べ、暗号鍵が盗まれる危険性を小さくできる。また、ある部分プログラムに関する暗号鍵が万一盗まれても、他の部分プログラムには影響しない。さらに、各部分プログラムが、プログラム本来の正しい処理の流れの中で正当なコール元から呼び出された場合にしか復号できないようにすることができるので、不正に制御を奪った者が不正参照の目的で特定の部分プログラムをメモリにロードさせようとしても、暗号鍵が得られないので不正参照は不可能である。

【００９１】

＜変形例４の備考＞

なお、上記の説明では、ある部分プログラムの復号用暗号鍵を得るためのデータ又はプログラムがコール元の部分プログラムに埋め込まれていることとしているが、それ以外の場所に埋め込む方法も考えられる。例えば、コールのタイミングで共有メモリ上に存在する部分プログラム（コール元以外のもの）のいずれかに暗号鍵取得のためのデータ又はプログラムを埋め込んでおいてもよい。また、ある部分プログラムの実行が終って共有メモリから消去されるタイミングで、そこに埋め込まれていた暗号鍵取得のためのデータ又はプログラムのみ制御部が読み出して専用エリアに保存しておき、その後実行される部分プログラムの復号に使用する、という方式も考えられる。

【００９２】

なお、上記の説明では、暗号鍵を得るためのデータ又はプログラムが単一の部分プログラム（コール元）に埋め込まれていることとしているが、複数の部分プログラムに分けて埋め込んでおいてもよい。

例えば、暗号鍵自体を埋め込む場合、復号対象の部分プログラムがコールされる時点で共有メモリＭにある複数の部分プログラムに暗号鍵を分けて埋め込んでおいて、復号時には、部分プログラム復号部がこれらを読み出し、組み合わせて暗号鍵を得る。どの部分プログラムのどの位置から読み出すか、読み出した複数の分割暗号鍵をどのような順序で組み合わせて暗号鍵とするかを示す情報は、暗

号化処理時に、例えばメモリ配置定義部が別途設定しておく。

【００９３】

暗号鍵取得プログラムを使う場合、例えば、部分プログラムＡ－Ｃ－Ｈの順で実行される際の部分プログラムＨ用の暗号鍵は、以下の手順で取得することもできる。まず、部分プログラムＡ実行時、これに埋め込まれた暗号鍵取得プログラムは、部分プログラムＨ用の暗号鍵の前半部分を、保持部１６内の暗号鍵格納用エリアの前半に書き込み、同暗号鍵の後半部分は部分プログラムＣ実行時、これに埋め込まれた暗号鍵取得プログラムによって暗号鍵格納用エリアの後半に書き込まれる。これによって、部分プログラムＨコール時には、部分プログラムＨ用の暗号鍵は保持部に完成した形で保持されている。

【００９４】

なお、以上に述べた複数の方式については、矛盾が生じない限りにおいて組み合わせた形で実施することも可能である。

また、上の説明では、暗号化プログラムに暗号鍵を埋め込む処理を行う構成部、鍵生成部９０１ａ、鍵取得プログラム生成部１２０１を暗号化プログラム生成部装置の一部としているが、メモリ配置定義部と同様、プログラムローダの一部としてもよい。

【００９５】

また、本変形例の場合、暗号化プログラム生成部装置の一部である、暗号化プログラムに暗号鍵を埋め込む処理を行う構成部、鍵生成部９０１ａ、鍵取得プログラム生成部１２０１とプログラムローダとを合わせたものが、本発明の暗号化データ復号装置を構成している。

（変形例５）

本変形例は、部分プログラムの暗号化／復号に用いる暗号鍵を鍵用の暗号鍵で暗号化しておくことで、機密保護を強化するというものである。さらに、鍵用暗号鍵は、各部分プログラム毎に個別のものを、プログラム暗号化処理に先立って生成しておく。これら鍵用暗号鍵は、それぞれが対応する部分プログラムからコールされる部分プログラムの暗号化／復号の際に暗号鍵の暗号化／復号に使用される。

【００９６】

図１３は、本変形例でのプログラムローダ１ｂと関連する装置の構成を示すブロック図である。実施の形態における構成に、個別暗号鍵生成部１３０１（暗号化プログラム生成装置Ｃ１３側）と暗号鍵復号部１３０２（プログラムローダ１ｂ側）とが追加された構成である。

・暗号化時の処理

暗号化プログラム生成装置Ｃ１３が、１つの部分プログラムを暗号化する処理を以下に説明する。先ず、暗号化プログラム生成装置Ｃ１３は、個別暗号鍵生成部１３０１に指示して、各部分プログラムに個別の鍵用暗号鍵を生成させる。そしてこれら鍵用暗号鍵を各部分プログラムに埋め込み、それと同時に埋め込み位置を示す情報（「埋め込み位置情報」）を生成する。

【００９７】

次いで、暗号化プログラム生成装置Ｃ１３は、部分プログラムを、プログラム暗号化用の暗号鍵（全部分プログラムに共通のもの）を用いて暗号化する。その後、変形例５で用いたのと同じ呼出関係情報を参照して、当該部分プログラムを呼び出すコール元部分プログラムを検出する。

そして、当該コール元部分プログラムに対応する鍵用暗号鍵を用いて、部分プログラム暗号化用の暗号鍵を暗号化する。そして、暗号化した暗号鍵を、対応する部分プログラム及びコール元部分プログラムの識別情報、そして、先に生成しておいた「埋め込み位置情報」と共に保持部１６に送り、暗号化暗号鍵用の領域に保持させる。

・復号における処理

復号処理は、暗号鍵復号部１３０２と部分プログラム復号部１３２３とで行う。

【００９８】

ある部分プログラムを復号しようとする部分プログラム復号部１３２３は先ず、暗号鍵復号部１３０２に対し、復号対象の部分プログラム及び、そのコール元部分プログラムの識別情報（制御部１１から取得するもの）を通知する。暗号鍵復号部１３０２は、これらを元に、保持部１６が各部分プログラムに対応して保

持している「暗号化された暗号鍵」及び「鍵用暗号鍵の埋め込み位置情報」の組の中から、当該復号対象およびコール元の部分プログラムの組み合わせに対応するものを読み出す。

【００９９】

ついで、暗号鍵復号部１３０２は、共有メモリM上の部分プログラムのうち、上記埋め込み位置情報が示すコール元部分プログラムの所定位置から鍵用暗号鍵を読み出す。そして、この鍵用暗号鍵を用いて、復号対象の部分プログラム用の「暗号化暗号鍵」を復号して暗号鍵を得ると、これを部分プログラム復号部１３２３に出力する。

【０１００】

部分プログラム復号部１３２３は、復号された暗号鍵を用いて当該部分プログラムを復号する。

なお、上の説明では、部分プログラム暗号化処理に用いる暗号鍵を共通とし、その暗号化する鍵用暗号鍵のみを部分プログラム毎に別々のものとしたが、暗号鍵自体についても部分プログラム毎に個別の内容とすることにしてもよい。

【０１０１】

また、上の説明では、個別暗号鍵生成部１３０１、埋め込み位置情報を生成する構成部、暗号化暗号鍵や埋め込み位置情報を保持部１６に格納する処理を行う構成部を暗号化プログラム生成部装置の一部としているが、メモリ配置定義部と同様、プログラムローダの一部としてもよい。

また、本変形例の場合、暗号化プログラム生成部装置の一部である、個別暗号鍵生成部１３０１、埋め込み位置情報を生成する構成部、暗号化暗号鍵や埋め込み位置情報を保持部１６に格納する処理を行う構成部と、プログラムローダとを合わせたものが、本発明の暗号化データ復号装置を構成している。

（備考）

以下、これまでに記した実施の形態、及び変形例に関して、他に留意すべきことがらを述べる。

【０１０２】

上で述べた実施の形態では、復号支援プログラムによって復号される側のプロ

グラムのみが部分プログラムに分割されて機密保護を実現しているが、復号支援プログラムについても、同様に部分プログラムの集合の形で暗号化しておき、部分プログラム単位で復号してメモリにロード、実行させることとしてもよい。そうすれば、復号処理方式のアルゴリズムや暗号鍵の内容が、メモリ上の復号支援プログラムから読み取られてしまう危険性を低減できるので、より高度な機密保護が実現できる。

【０１０３】

また、上で述べた実施の形態の処理に加え、各部分プログラムについて、メモリＭへのロード時点と実行完了時点とでその内容を比較することで、当該部分プログラムがメモリＭへのロード後、実行中に改変されていないかをチェックすることとしてもよい。そして、改変を検出した場合は、処理を打ち切って復号済みデータを消去する。

【０１０４】

また、上記の暗号鍵は、具体的には例えばＤＥＳのような暗号化方式に用いる暗号鍵であり、通常、プログラムの一定領域に埋め込まれるかあるいはユーザには見えない部分あるいはファイルに秘匿されている、としているが、暗号方式はこれ以外のものでもよい。簡易化する方法として、単に値の排他的論理和を取る方法等でもよい。また秘匿の方法も上記のものに限定されない。

【０１０５】

また、部分プログラム復号部、格納アドレス復号部、メモリ配置情報復号部は、まとめて単一の構成としてもよいし個別の構成としてもよい。単一の構成とする場合、復号処理の要求元が復号対象のデータの種類を示す情報をパラメータとして設定し、復号部はこれを参照してデータの種類に応じた復号処理を行うこととする。

【０１０６】

また、プログラムを復号処理の対象として説明したが、プログラム以外のデータであってもよい。

また、部分プログラムについては、元となるプログラムをモジュールやルーチンの単位で分割することにより生成されるものとしてもよいが、ＤＬＬなどのよ

うに予め用意されている部品を組み合わせて単一のプログラムとして動作させる場合は、こうした部品を単独で、又は組み合わせて部分プログラムとしてもよい。本発明の前提は複数の部分プログラムが集合して単一のプログラムとして動作する、ということであり、部分プログラムがどのように生成されるかは本質的問題ではない。

【０１０７】

また、実施の形態、及び変形例で示した各装置は、コンピュータによって実行されるプログラムの形で実現することもできる。そして、プログラムとして実現する場合は、その暗号化データ復号プログラム自身も暗号化部分プログラムの集合の状態で保存しておき、部分プログラム単位で復号のうえ実行する形としてもよい。

【０１０８】

【発明の効果】

以上の説明から明らかなように、本発明の暗号化データ復号装置は、暗号化された形で保存されているプログラムをコンピュータシステム上での実行のために復号する暗号化データ復号装置であって、プログラムを暗号化された複数の部分プログラムの集合の形で保持している記憶手段と、前記部分プログラムの各々について、復号のタイミングを示すタイミング情報と、復号後の前記実行用メモリ空間における配置エリアを示す位置情報とからなるメモリ配置情報を、前記プログラムの暗号化の際に予め生成しておくメモリ配置情報生成手段と、前記記憶手段から、前記タイミング情報に従って部分プログラムを順次読み出して復号する復号手段と、前記復号手段によって復号された部分プログラムを、前記位置情報に従って前記実行用メモリ空間内の配置エリアに配置するロード手段と、を有する、という構成を特徴とする。

【０１０９】

このような暗号化データ復号装置によれば、実行用メモリに配置されるのはプログラムの一部のみであるため、割り込みなどを利用してメモリ上のデータを不正に参照される事態になったとしても、被害は最小限に抑制できる。また、前記メモリ配置情報における位置情報について、前記複数の部分プログラムの少なく

とも一部について、当該部分プログラムの復号より前に別の部分プログラムが配置されていた配置エリアに上書きされるような内容として生成すると、特定の部分プログラムが実行用メモリ上に存在する時間が短くなり、その分、不正参照されにくくなる。

【０１１０】

また、前記メモリ配置情報のうち前記位置情報は暗号化された状態で保持されており、前記ロード手段は、前記位置情報を復号し、復号後の位置情報が示す配置エリアに復号後の部分プログラムを配置すること、としてもよい。この構成によれば、メモリ配置情報が不正参照されることで復号後の部分プログラムのロード位置が知られるという事態を防止できるので、メモリロード後のプログラムの機密性を高めることができる。

【０１１１】

また、前記ロード手段は、配置エリアに部分プログラムをロードする際、当該配置エリアのサイズが当該部分プログラムのサイズより大きければ、サイズ差によって生じる当該配置エリアの空き領域にダミーデータを書き込むこと、としてもよい。この構成によれば、部分プログラムのうち、上書きによって消去されない部分についても、長時間メモリ上に残される事態を防止できるので、この部分が不正参照される危険性は低くなる。

【０１１２】

また、前記ロード手段は、ある配置エリアにロードされた部分プログラムの実行が終了した時点から所定時間が経過しても当該配置エリアへの次の部分プログラムの配置が行われない場合、当該配置エリア内に配置されている部分プログラムを消去すること、としてもよい。この構成によれば、処理の終了した部分プログラムが長時間メモリ上に残される事態を防止できるので、この部分プログラムが不正参照される危険性は低くなる。

【０１１３】

また、暗号化処理の際に予め、前記複数の部分プログラムの少なくとも一部に、他の部分プログラムの復号処理に用いられる暗号鍵の一部又は全部を埋め込んでおく鍵埋め込み手段をさらに有し、前記復号手段は、部分プログラムを復号す

る場合、それ以前に復号されて前記実行用メモリ空間に配置されている１個以上の他の部分プログラムに埋め込まれた暗号鍵を取得して当該復号対象の部分プログラムを復号すること、としてもよい。この構成によれば、不正利用者がシステムの制御を奪って、特定の部分プログラムを不正参照の目的でメモリ上にロードさせようとしても、当該部分プログラムの復号に必要な暗号鍵を有する他の部分プログラムがメモリ上に存在しないため、その不正参照はできない。

なお、同様の効果を得るために、暗号化処理の際に予め、前記複数の部分プログラムの少なくとも一部に、他の部分プログラムの復号処理に用いられる暗号鍵の全部又は一部を生成するための暗号鍵生成プログラムを埋め込んでおくプログラム埋め込み手段をさらに有し、前記復号手段は、部分プログラムを復号する場合、それ以前に復号されて前記実行用メモリ空間に配置されている１個以上の他の部分プログラムに埋め込まれていた暗号鍵生成プログラムの実行によって生成された暗号鍵を用いて当該復号対象の部分プログラムを復号すること、または、暗号化処理の際に予め、前記複数の部分プログラムの少なくとも一部に、他の部分プログラムの復号処理に用いられる暗号鍵を復号するための鍵用暗号鍵を埋め込んでおく鍵用暗号鍵埋め込み手段をさらに有し、前記復号手段は、部分プログラムを復号する場合、それ以前に復号されて前記実行用メモリ空間に配置されている他の部分プログラムから取得した鍵用暗号鍵を用いて復号した暗号鍵によって当該復号対象の部分プログラムを復号すること、という構成も考えられる。

【０１１４】

また、前記ロード手段は、最初に復号される部分プログラムの実行用メモリ空間への配置に先立って前記配置エリアの絶対アドレスを動的に決定すること、としてもよい。この構成によれば、プログラムのメモリ上ロード位置が実行のたびに変わるので、不正利用者がロード位置を推測しながらメモリの内容を観察することでプログラムの内容を不正参照しようとしても、成功する可能性は低い。

【０１１５】

また、前記復号手段による復号処理は復号支援プログラムを用いて行われ、前記復号支援プログラムの正当性を確認する復号プログラム確認手段を更に有し、前記復号手段は、部分プログラムの復号に先立って、前記復号支援プログラム認

証部に前記復号支援プログラムの正当性確認を行わせ、正当性が認証された場合にのみ部分プログラムの復号を行うこと、としてもよい。この構成によれば、復号支援プログラムを悪用してプログラムの内容を不正参照することは不可能であり、復号処理時の機密性を向上させることができる。

【０１１６】

また、割込みが検知された場合に、不正アクセス防止処理として、前記実行用メモリ空間上に配置済みの部分プログラムを消去する不正アクセス防止手段を更に有すること、としてもよい。この構成によれば、割り込みを悪用してプログラムの内容を不正参照することは不可能となる。これに加え、前記不正アクセス防止手段は、前記不正アクセス防止処理実行の際に、ダミープログラムを実行させること、とすることもできる。また、前記不正アクセス防止手段は、正当なプログラム逆解析のための割込み発生位置の登録を予め受け付けておき、検知された割込みが当該登録された割込み発生位置で発生していた場合は、前記不正アクセス防止処理を実行しないこと、とすれば、不正参照防止のために正当な割り込みまでが禁止されるという不都合は生じない。

【０１１７】

また、前記部分プログラムの各々について、前記記憶手段における格納位置を示す格納位置情報を暗号化した状態で保持する格納位置情報保持部を更に有し、前記復号手段は、前記格納位置情報保持部から読み出して復号した格納位置情報に従って、前記記憶手段から部分プログラムを読み出し、復号すること、としてもよい。この構成によれば、正当な部分プログラムだけが実施される。すなわち、不正利用者が記憶装置に格納しておいた不正参照用のプログラムを正当な部分プログラムの代わりに実行させる、という形での不正参照を防止できる。これに加え、前記格納位置情報が正当なものか否かを判定する格納位置情報認証部を更に有し、前記復号手段は、前記格納位置情報認証部によって格納位置情報が正当と判定された場合に、前記記憶手段から部分プログラムを読み出し、復号すること、としてもよい。

【０１１８】

また、上に述べた効果は、上記の特徴を有する暗号化データ復号装置が実行す

る暗号化データ復号方法や、当該復号方法をコンピュータに実行させるプログラムによっても達成することができる。また、上記の特徴を有する暗号化データ復号装置によって利用されるメモリ配置情報、暗号鍵、暗号鍵生成プログラム、鍵用暗号鍵を生成しながらプログラムの暗号化を行う暗号化プログラム生成装置も、上記の効果を得る上で有用である。

【図面の簡単な説明】

【図１】　本発明に関わる暗号化データ復号装置の１実施の形態であるプログラムローダの構成を、これが動作するコンピュータシステムにおいて示すブロック図である。

【図２】　同実施の形態における格納アドレス情報とそれが示す部分プログラムの２次記憶装置上の格納位置との対応を示すイメージ図である。

【図３】　同実施の形態におけるメモリ配置情報の構成と内容の一例とを示す概念図である。

【図４】　同実施の形態において復号された部分プログラムが共有メモリ上にどう配置されるかを示した概念図である。

【図５】　同実施の形態における暗号化プログラムの復号及びロード処理におけるプログラムローダの動作を示すフローチャートである。

【図６】　エリアアドレス情報の例を示す図である。

【図７】　別のエリアアドレス情報の例を示す図である。

【図８】　ダミーデータ埋め込み処理の概要を時間経過に沿って示すイメージ図である。

【図９】　ある変形例におけるプログラムローダの構成を示すブロック図である。

【図１０】　呼出関係情報によって定義される部分プログラム間の呼出関係を示す模式図である。

【図１１】　復号処理時の部分プログラムのメモリ配置を時間軸に沿って示す模式図である。

【図１２】　ある変形例におけるプログラムローダの構成を示すブロック図である。

【図１３】　ある変形例におけるプログラムローダの構成を示すブロック図である。

【符号の説明】

１、１ｂ、９、９ａ　プログラムローダ

１１、９１１　制御部

１２　復号部

１２１　格納アドレス復号部

１２２　メモリ配置情報復号部

１２３、９２３、１２２３、１３２３　部分プログラム復号部

１３　復号支援プログラム認証部

１４　メモリ配置定義部

１５　不正アクセス防止部

１６　保持部

１７　格納アドレス情報認証部

９０１ａ　鍵生成部

１２０１　鍵取得プログラム生成部

１３０１　個別暗号鍵生成部

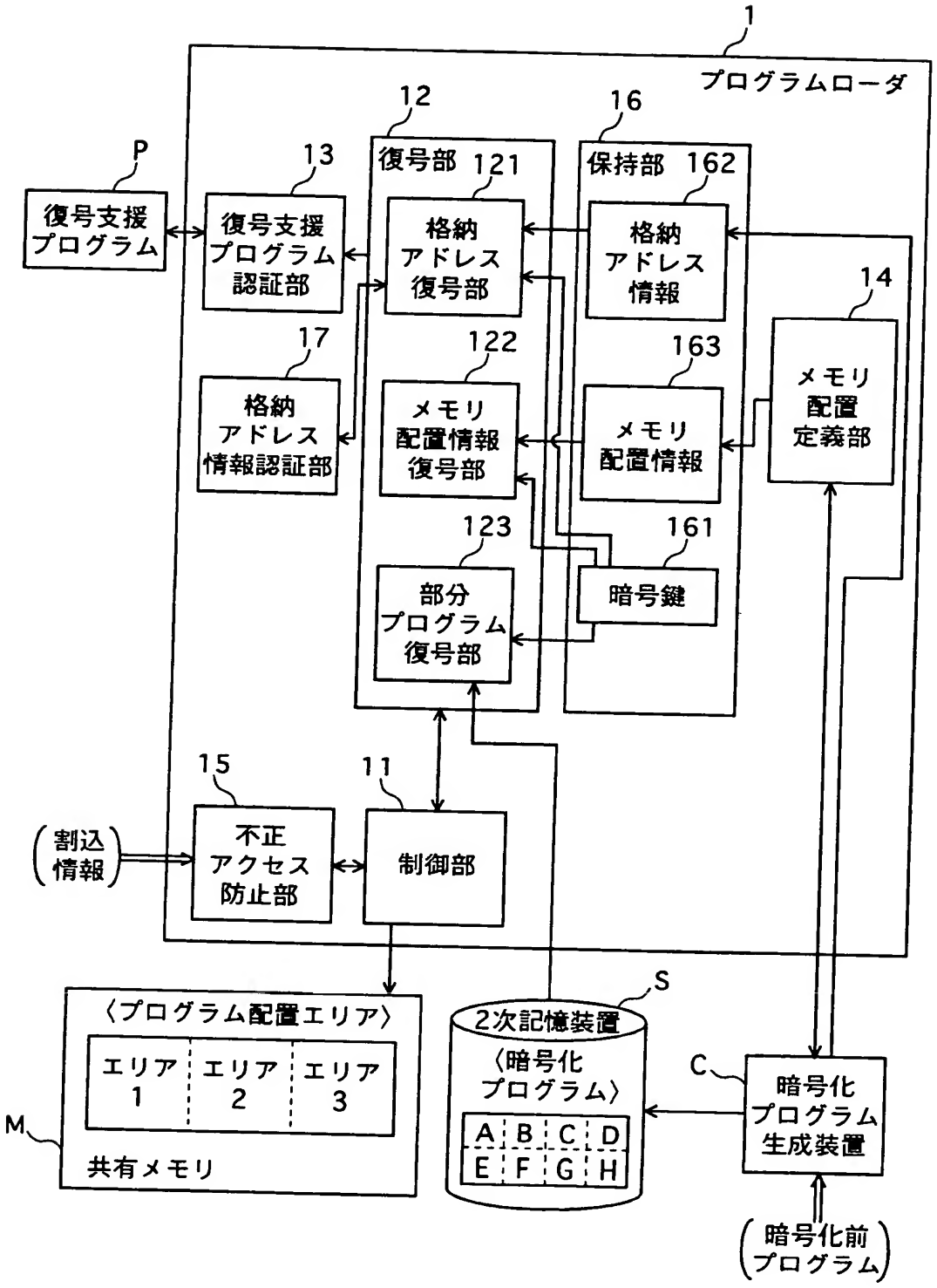
Ｃ、Ｃ１２、Ｃ１３　暗号化プログラム生成装置

Ｍ　共有メモリ

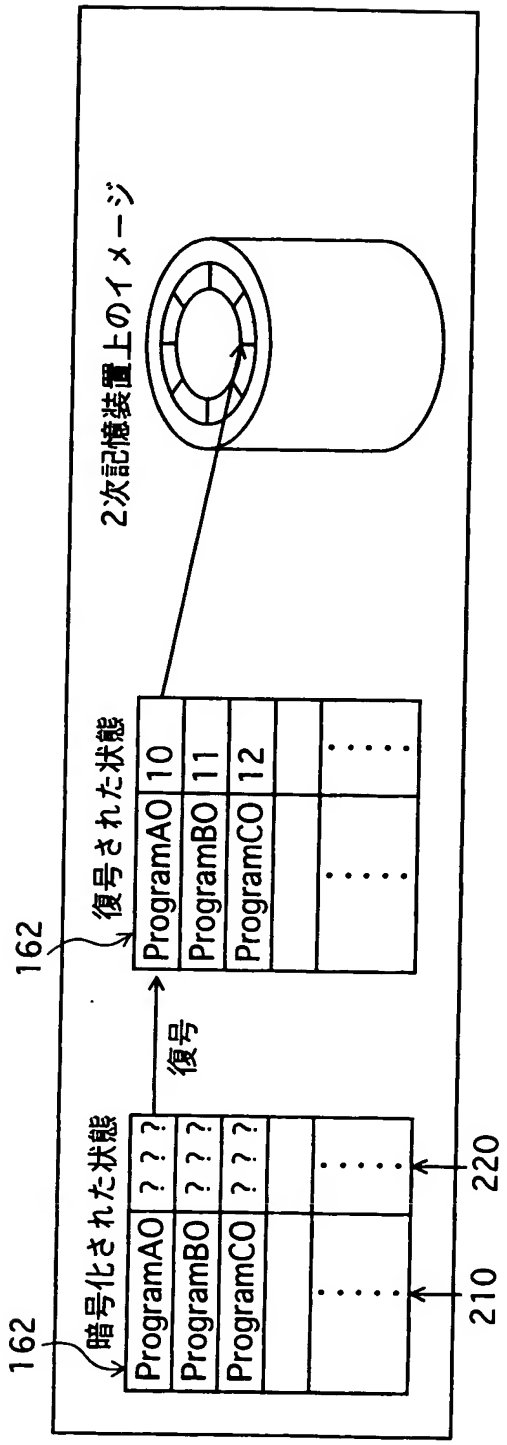
Ｓ　２次記憶装置

【書類名】　図面

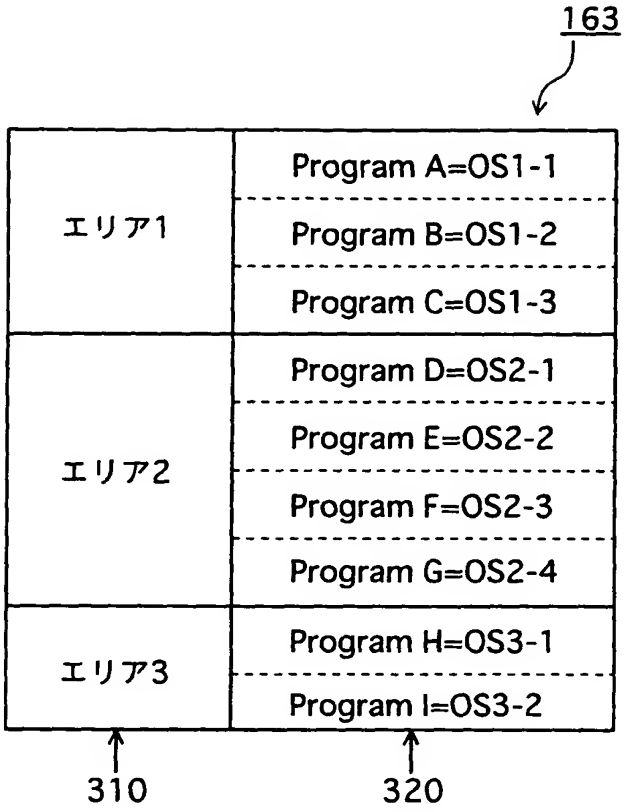
【図１】


1
プログラムローダ
12
16
P
13
復号部
121
保持部
162
復号支援
プログラム
復号支援
プログラム
認証部
格納
アドレス
復号部
格納
アドレス
情報
14
17
122
163
格納
アドレス
情報認証部
メモリ
配置情報
復号部
メモリ
配置情報
メモリ
配置
定義部
123
161
部分
プログラム
復号部
暗号鍵
15
11
割込
情報
不正
アクセス
防止部
制御部
S
〈プログラム配置エリア〉
エリア
1
エリア
2
エリア
3
2次記憶装置
〈暗号化
プログラム〉
A
B
C
D
E
F
G
H
C
暗号化
プログラム
生成装置
M
共有メモリ
暗号化前
プログラム

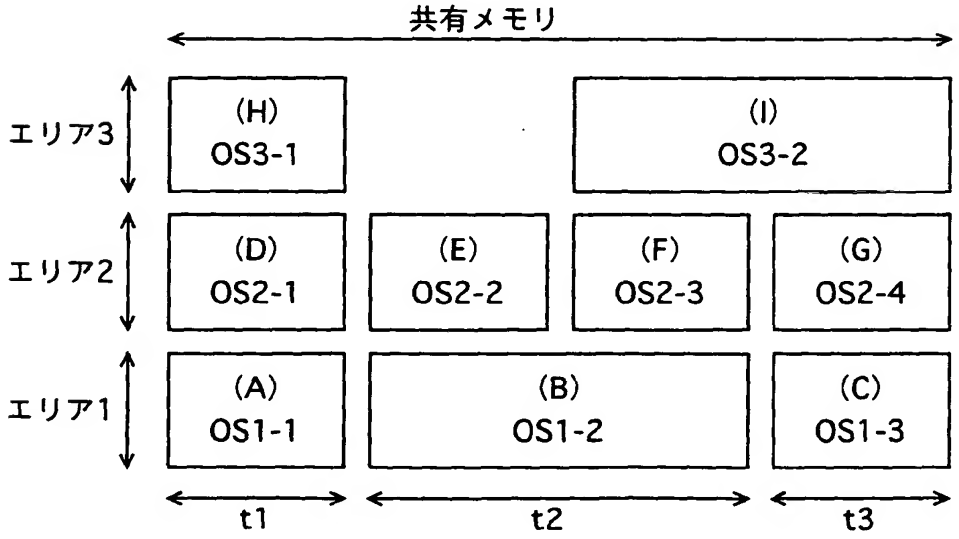
【図２】


2次記憶装置上のイメージ
復号された状態
ProgramA0 10
ProgramB0 11
ProgramC0 12
162
復号
暗号化された状態
ProgramA0 ? ? ?
ProgramB0 ? ? ?
ProgramC0 ? ? ?
162
220
210

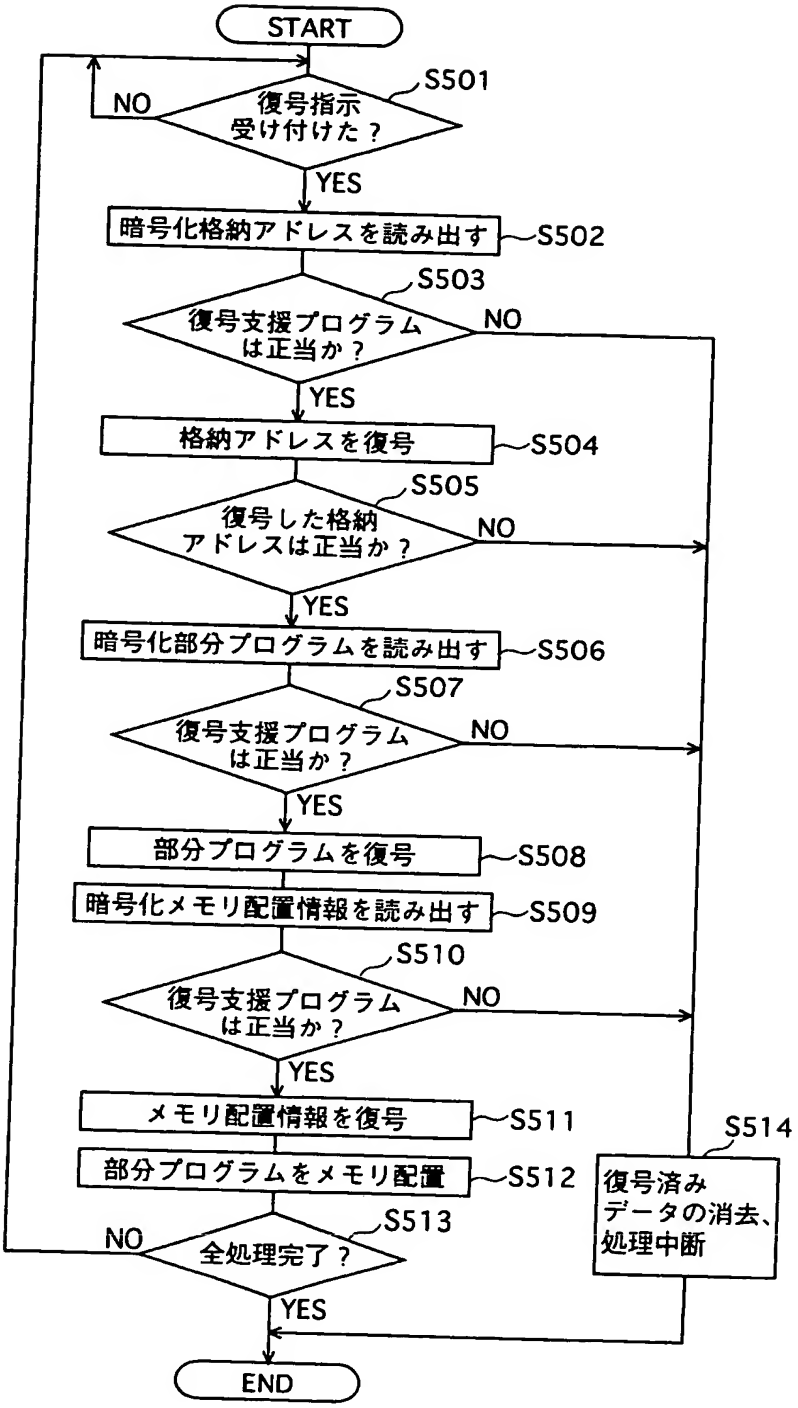
【図３】


163
エリア1
Program A=OS1-1
Program B=OS1-2
Program C=OS1-3
エリア2
Program D=OS2-1
Program E=OS2-2
Program F=OS2-3
Program G=OS2-4
エリア3
Program H=OS3-1
Program I=OS3-2
310
320

【図４】

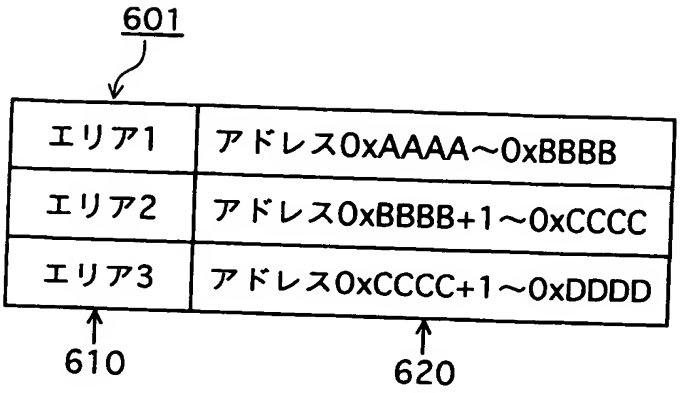

共有メモリ
エリア3
(H)
OS3-1
(I)
OS3-2
エリア2
(D)
OS2-1
(E)
OS2-2
(F)
OS2-3
(G)
OS2-4
エリア1
(A)
OS1-1
(B)
OS1-2
(C)
OS1-3
t1
t2
t3

【図５】

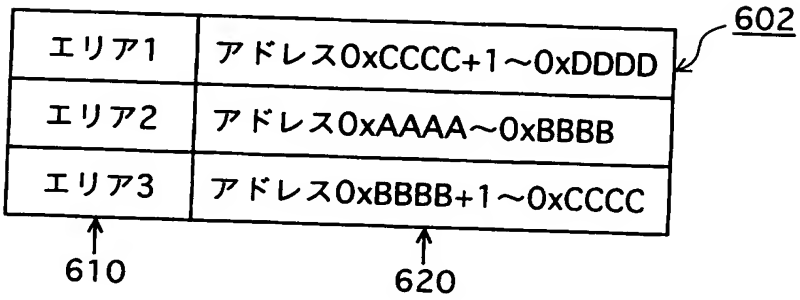

START
S501
復号指示
受け付けた？
NO
YES
暗号化格納アドレスを読み出す
S502
S503
復号支援プログラム
は正当か？
NO
YES
格納アドレスを復号
S504
S505
復号した格納
アドレスは正当か？
NO
YES
暗号化部分プログラムを読み出す
S506
S507
復号支援プログラム
は正当か？
NO
YES
部分プログラムを復号
S508
暗号化メモリ配置情報を読み出す
S509
S510
復号支援プログラム
は正当か？
NO
YES
メモリ配置情報を復号
S511
部分プログラムをメモリ配置
S512
S513
全処理完了？
NO
YES
S514
復号済み
データの消去、
処理中断
END

【図６】

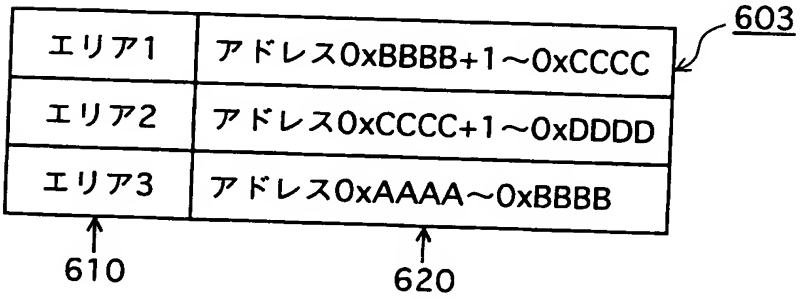
(1)


601
エリア1
アドレス0xAAAA～0xBBBB
エリア2
アドレス0xBBBB+1～0xCCCC
エリア3
アドレス0xCCCC+1～0xDDDD
610
620

(2)


602
エリア1
アドレス0xCCCC+1～0xDDDD
エリア2
アドレス0xAAAA～0xBBBB
エリア3
アドレス0xBBBB+1～0xCCCC
610
620

(3)


603
エリア1
アドレス0xBBBB+1～0xCCCC
エリア2
アドレス0xCCCC+1～0xDDDD
エリア3
アドレス0xAAAA～0xBBBB
610
620

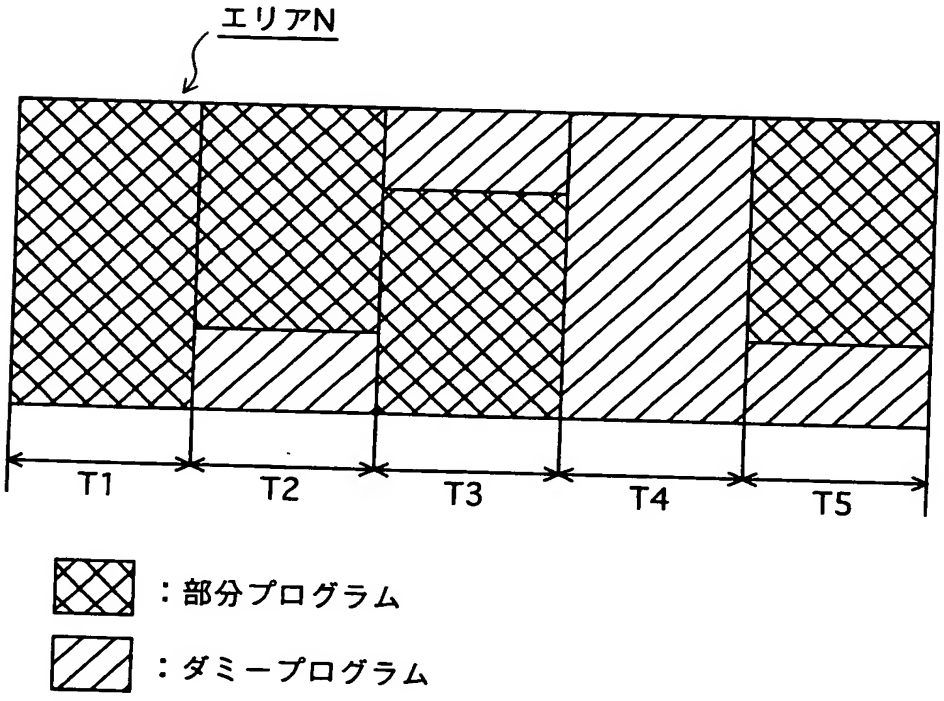
【図７】

700

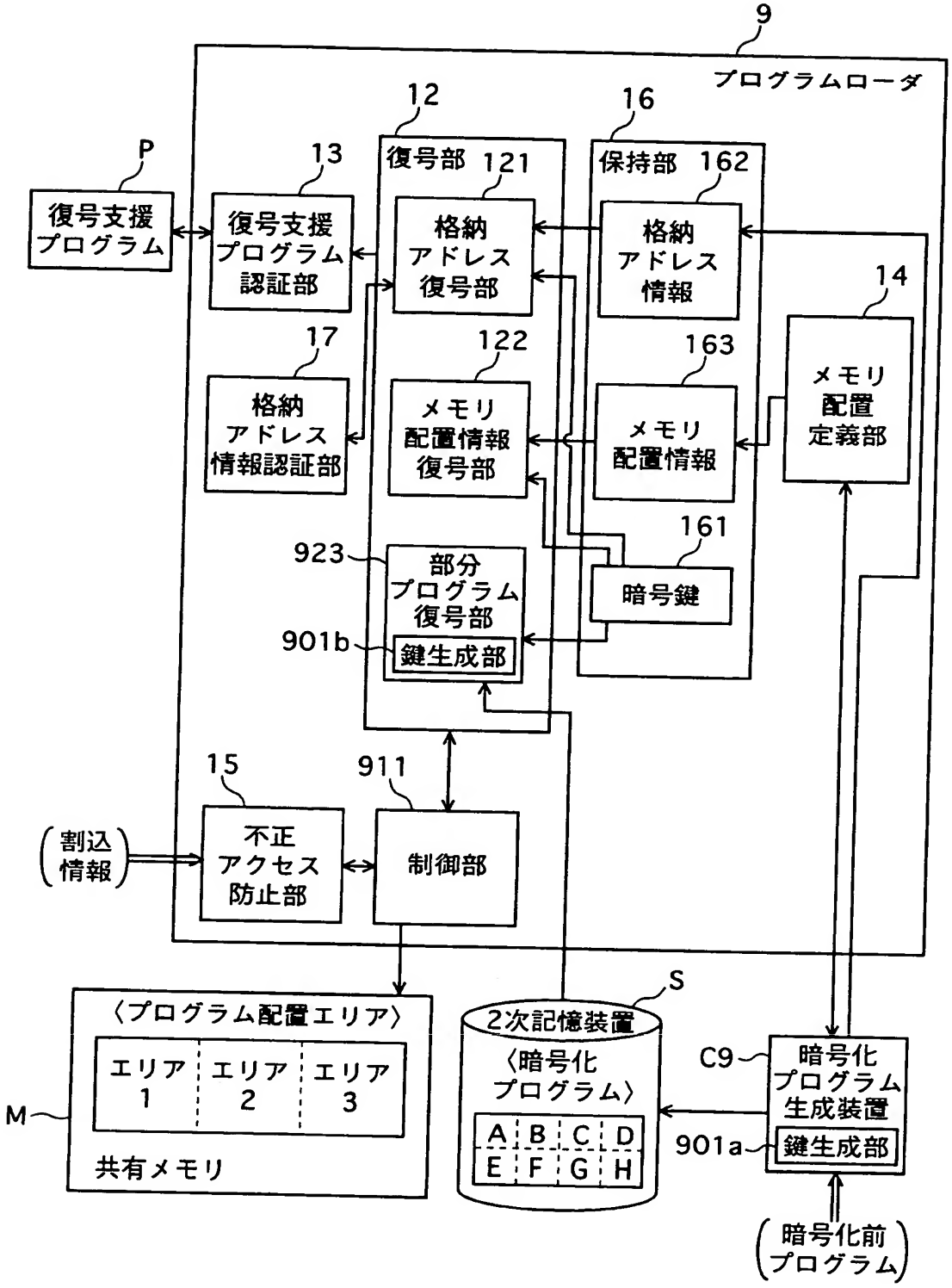
エリア1	アドレス0xAAAA+α～0xBBBB
エリア2	アドレス0xBBBB+1+β～0xCCCC
エリア3	アドレス0xCCCC+1+γ～0xDDDD

$$\begin{cases} 0 < \alpha + \text{size} < 0\text{xBBBB} \\ 0 < \beta + \text{size} < 0\text{xCCCC} \\ 0 < \gamma + \text{size} < 0\text{xDDDD} \end{cases}$$

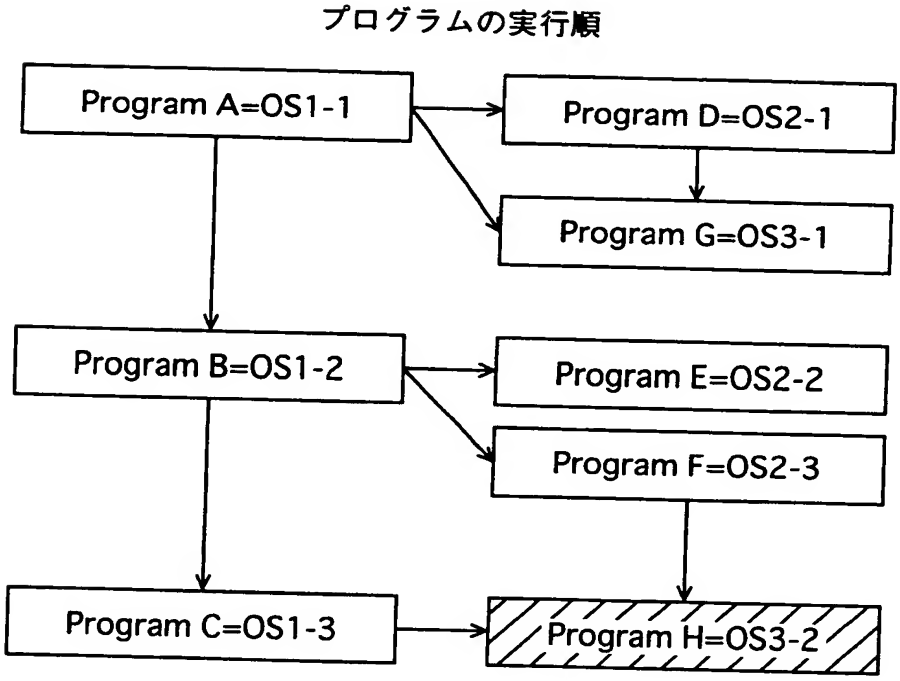
【図８】



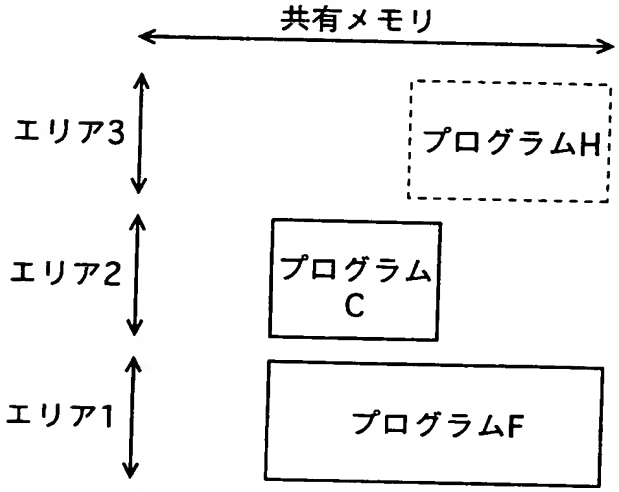
【図９】


9
プログラムローダ
12
復号部
121
格納
アドレス
復号部
122
メモリ
配置情報
復号部
923
部分
プログラム
復号部
901b
鍵生成部
16
保持部
162
格納
アドレス
情報
163
メモリ
配置情報
161
暗号鍵
14
メモリ
配置
定義部
P
復号支援
プログラム
13
復号支援
プログラム
認証部
17
格納
アドレス
情報認証部
15
不正
アクセス
防止部
911
制御部
割込
情報
〈プログラム配置エリア〉
エリア
1
エリア
2
エリア
3
M
共有メモリ
S
2次記憶装置
〈暗号化
プログラム〉
A
B
C
D
E
F
G
H
C9
暗号化
プログラム
生成装置
901a
鍵生成部
暗号化前
プログラム

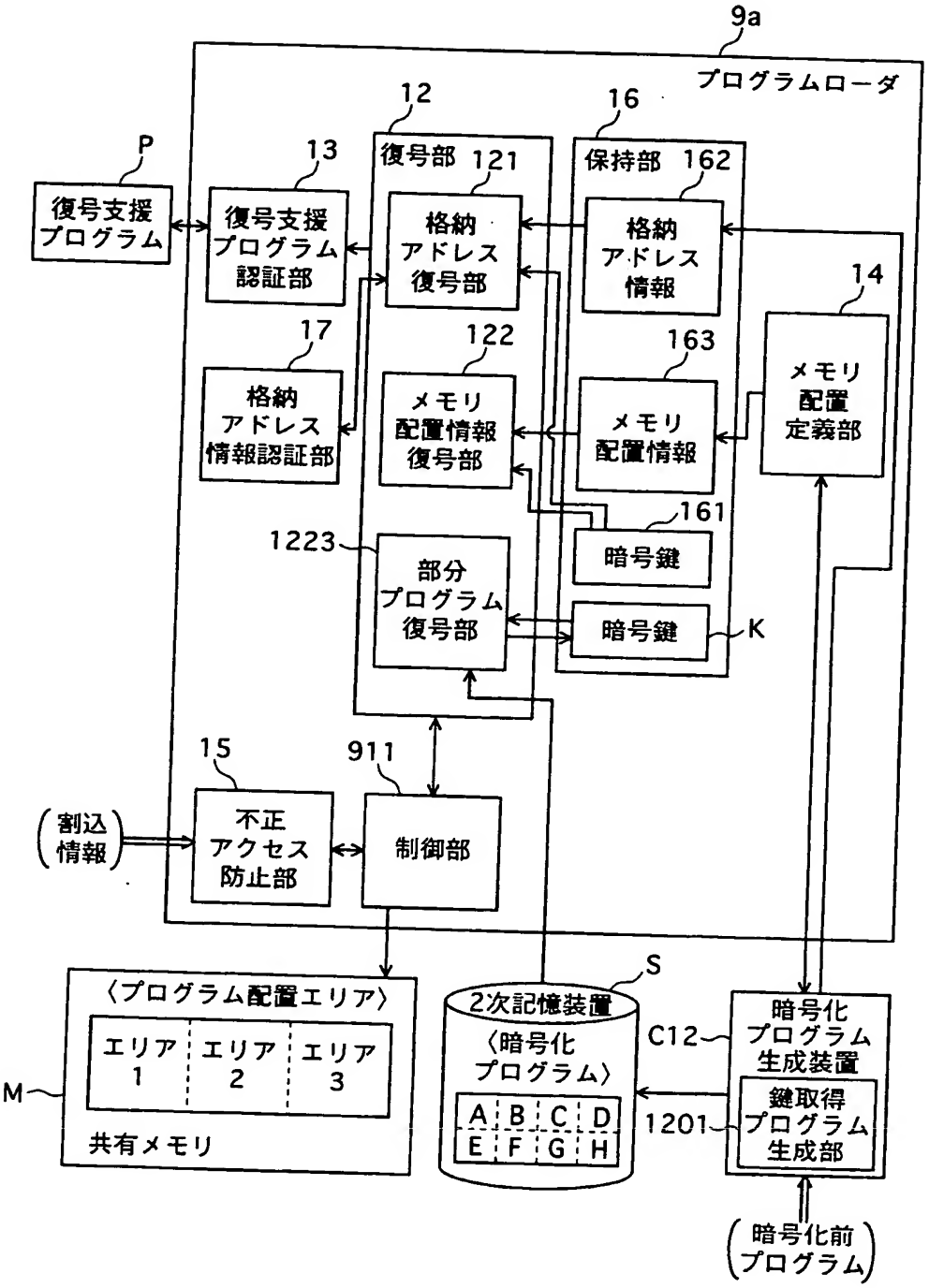
【図１０】


プログラムの実行順
Program A=OS1-1
Program D=OS2-1
Program G=OS3-1
Program B=OS1-2
Program E=OS2-2
Program F=OS2-3
Program C=OS1-3
Program H=OS3-2

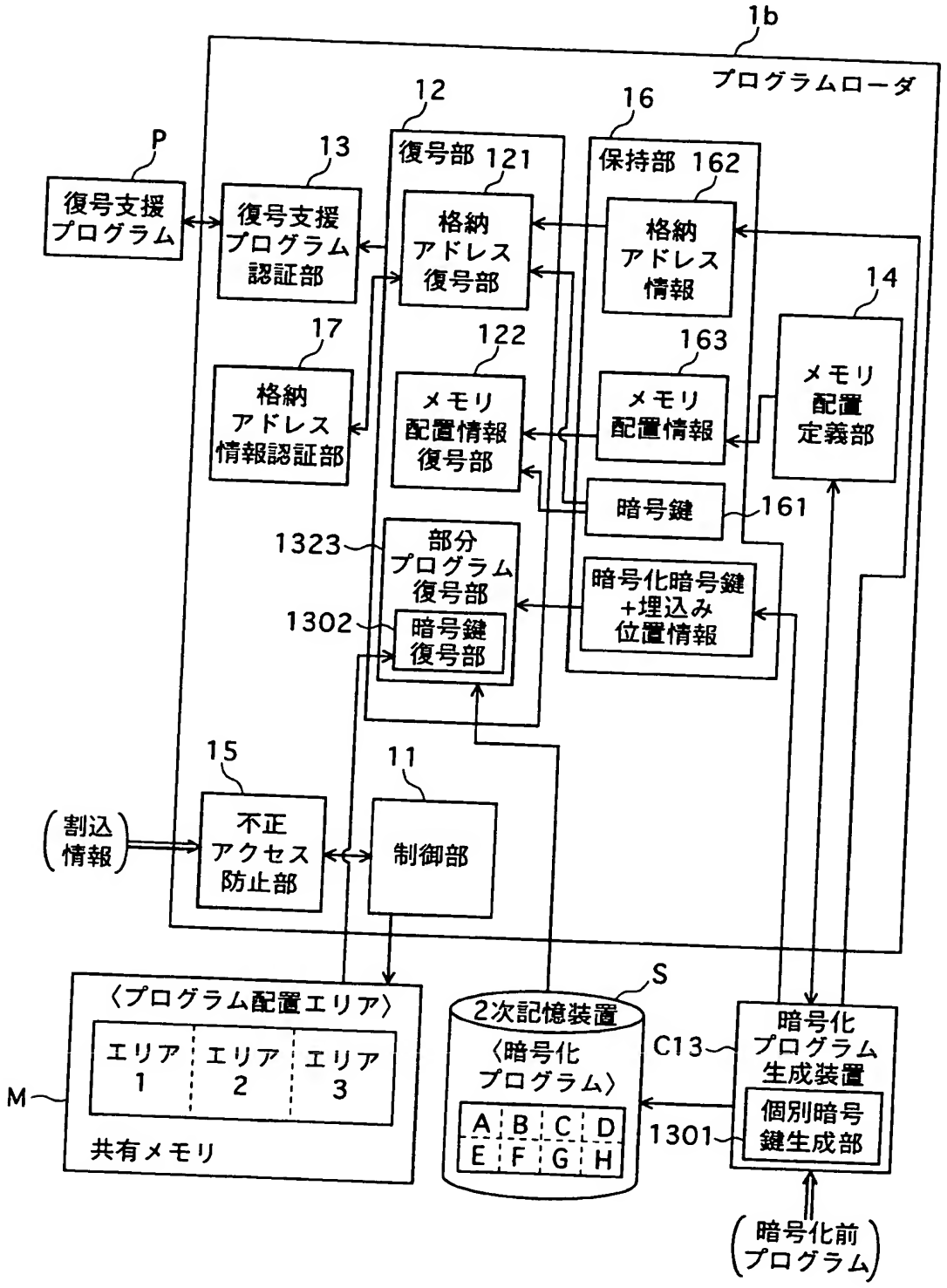
【図１１】


共有メモリ
エリア3
プログラムH
エリア2
プログラム
C
エリア1
プログラムF

【図１２】


9a
プログラムローダ
12
復号部
121
格納アドレス復号部
122
メモリ配置情報復号部
1223
部分プログラム復号部
16
保持部
162
格納アドレス情報
163
メモリ配置情報
161
暗号鍵
暗号鍵
K
14
メモリ配置定義部
P
復号支援プログラム
13
復号支援プログラム認証部
17
格納アドレス情報認証部
15
不正アクセス防止部
911
制御部
割込情報
〈プログラム配置エリア〉
エリア1
エリア2
エリア3
共有メモリ
M
2次記憶装置
〈暗号化プログラム〉
A B C D
E F G H
S
暗号化プログラム生成装置
C12
鍵取得プログラム生成部
1201
暗号化前プログラム

【図１３】


1b
プログラムローダ
12
復号部 121
16
保持部 162
P
復号支援プログラム
13
復号支援プログラム認証部
格納アドレス復号部
格納アドレス情報
14
メモリ配置定義部
17
格納アドレス情報認証部
122
メモリ配置情報復号部
163
メモリ配置情報
暗号鍵
161
1323
部分プログラム復号部
暗号化暗号鍵+埋込み位置情報
1302
暗号鍵復号部
15
11
割込情報
不正アクセス防止部
制御部
S
〈プログラム配置エリア〉
エリア1
エリア2
エリア3
M
共有メモリ
2次記憶装置
〈暗号化プログラム〉
A B C D
E F G H
C13
暗号化プログラム生成装置
個別暗号鍵生成部
1301
暗号化前プログラム

【書類名】　要約書

【要約】

【課題】　暗号化プログラムやデータの復号から実行の過程において、これらプログラムやデータの機密性を向上させることのできる暗号化データ復号装置を提供する。

【解決手段】　制御部１１は、復号された部分プログラムを、共有メモリＭにロードする際、メモリ配置情報が示すエリアにロードする。メモリ配置情報は、共通するエリアに順次部分プログラムが上書きされる形でロードされるように設定されているので、特定の部分プログラムがメモリ上に長時間存在することはなく、その分、不正参照されにくくなる。また、各種データの復号のたびに復号支援プログラム認証部１３が復号支援プログラムＰの認証を行うことで、復号支援プログラムを悪用した不正参照を防止する。

【選択図】　図１

出証特２００３－３０４８７２５

出 願 人 履 歴 情 報

識別番号　〔０００００５８２１〕

１．変更年月日　１９９０年　８月２８日
［変更理由］　新規登録
住　所　大阪府門真市大字門真１００６番地
氏　名　松下電器産業株式会社